

Trabajo Fin de Grado (TFG)



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE SISTEMAS
INFORMÁTICOS

GRADO EN INGENIERÍA DE SOFTWARE 2024-2025

Interfaz de un sistema de gestión de huella
de carbono para el sector vinícola

Autor:

Xinyao Alicia Zhang

Tutor:

Pilar Manzano García (Tutor académico)

Miguel Ángel López Peña (Tutor profesional)

Contenido

Contenido.....	2
1. Resumen	1
2. Abstract.....	2
3. Agradecimientos.....	3
4. Introducción.....	4
4.1. Objetivos del proyecto.....	4
4.2. Estructura del documento	5
5. Fundamentos teóricos y técnicos.....	6
5.1. Conceptos teóricos	6
5.2. Información técnica	7
5.3. Librerías y lenguajes.....	9
5.3.1. Desarrollo de frontend	9
5.3.2. Desarrollo de backend	10
5.3.3. Bases de datos	10
6. Estado del arte.....	11
6.1. Historia y necesidades	11
6.2. Soluciones existentes.....	11
7. VIDEXPERT	12
7.1. Metodología de trabajo	14
7.2. Aplicaciones externas de apoyo	15
8. Requisitos del módulo	17
8.1. Requisitos funcionales	17
8.2. Requisitos no funcionales	17
9. Modelo de datos.....	19
9.1. Ejemplos visuales de datos	26
10. Diseño del módulo	29
10.1. Diagrama de casos de uso	29
10.2. Librerías externas.....	33
10.3. Diagrama de clases	34
10.4. Diagrama de componentes de sistema	46
10.5. Diagramas de actividad.....	48
10.6. Comportamiento visual	55

10.6.1.	Componente survey.....	56
10.6.2.	Componente section.....	57
11.	Implementación.....	63
12.	Pruebas y Validación.....	66
12.1.	Pruebas unitarias.....	66
12.2.	Pruebas de interfaz gráfica.....	66
12.3.	Pruebas de aceptación del cliente.....	67
13.	Conclusión.....	68
13.1.	Resultados obtenidos.....	68
13.2.	Limitaciones y posibles mejoras.....	69
13.3.	Impacto.....	70
14.	Bibliografía.....	72

Índice de figuras

Figura 1: Microservicios de la aplicación VIDEXPERT	13
Figura 2: Ciclo del Scrum [21]	14
Figura 3: Product backlog del Jira	15
Figura 4: Tablas de estructuras de datos	20
Figura 5: Tipos de desplegables	25
Figura 6: Ejemplo de city, survey_year, date	26
Figura 7: Subsección de respuesta múltiple	26
Figura 8: Subsección con item, pero algunas preguntas no tienen item	27
Figura 9: Subsección con preguntas normales e ítems	27
Figura 10: Sección con subsection_election y la otra no	28
Figura 11: Diagrama de casos de uso	29
Figura 12: Diagrama de clases (<i>sections</i>)	35
Figura 13: Diagrama de clases (<i>survey</i>)	44
Figura 14: Diagrama de componentes	47
Figura 15: Diferencia entre “section sidebar” y “normal sidebar”	48
Figura 16: Actividad iniciar sección	49
Figura 17: Actividad cambiar sección/página	50
Figura 18: Actividad guardar respuestas	51
Figura 19: Actividad cargar Excel	52
Figura 20: Actividad de editar respuesta	53
Figura 21: Actividad de actualizar <i>question.option</i>	54
Figura 22: Cambiar el dropdown según el valor de otra pregunta	55
Figura 23: Flujo de actividad en frontend de survey	56
Figura 24: Partes de section o static	57
Figura 25: Página de solo lectura	58
Figura 26: Formas de cambiar de sección	59
Figura 27: Alerta de respuestas sin guardar	59
Figura 28: Alerta de subsección no guardado	60
Figura 29: Alerta de respuestas erróneas	60
Figura 30: Detalles de encuesta	60
Figura 31: Tutorial section	61
Figura 32: Atributo “ <i>subsection_election</i> ”	61
Figura 33: Carga para hacer cálculos	62
Figura 34: Comportamiento de hidden	62
Figura 35: Rutas anidadas	63
Figura 36: BehaviorSubject de data.control	64
Figura 37: Redirección en base al URL	64
Figura 38: Página de estilos de la aplicación style.scss	65
Figura 39: Ejemplo de tutorial (paso 1 de 10)	65
Figura 40: Botón de tutorial en navbar	65

1. Resumen

En los últimos años, el cambio climático junto con la creciente demanda de productos sostenibles en el mercado ha obligado a los productores de vino a reducir la huella de carbono en los procesos de la vinificación. Actualmente, las herramientas disponibles para medir y gestionar las emisiones de CO₂ presentan limitaciones en usabilidad, precisión y capacidad de análisis comparativo. VIDEXPERT surge como respuesta ante dicha necesidad: a través de los datos recolectados desde una interfaz web, calcula las emisiones de CO₂ producidas durante las actividades para la obtención de vino, desde la plantación de la uva hasta el embotellado del producto final. Mediante un sistema inteligente de diagnóstico y recomendaciones, VIDEXPERT ofrece a los usuarios unas pautas personalizadas que ayudan a reducir la huella de carbono.

Este Proyecto Fin de Grado tiene como objetivo principal diseñar e implementar el módulo de recolección de datos de VIDEXPERT. Este componente requiere de una interfaz gráfica de usuario intuitiva y amigable que facilita a los usuarios introducir, validar y almacenar datos sobre los procesos, métodos y productos usados durante los procesos de producción del vino para su posterior análisis y cálculo de la huella de carbono. Con la aplicación de VIDEXPERT se espera estandarizar la medición de emisiones de gases de efecto invernadero en el sector vitivinícola, facilitar adopción de prácticas sostenibles y optimizar los procesos de obtención del vino.

Durante el desarrollo de VIDEXPERT se ha seguido la metodología ágil Scrum, utilizando **Angular** como **Framework** para el frontend (HTML), **Spring Boot** para el backend y **MongoDB** como base de datos.

Como futura mejora, se propone incorporar inteligencia artificial para optimizar las recomendaciones de mitigación de la huella de carbono.

2. Abstract

In recent years, climate change and the growing demand for sustainable products in the market has forced wine producers to reduce the carbon footprint in winemaking processes. Currently, the tools available to measure CO₂ emissions present limitations in usability, accuracy and comparative analysis capacity. VIDEXPERT was developed as a response to this need: through data collected from a web interface, it calculates the CO₂ emissions produced during the activities to obtain wine (from the planting of the grapes to the bottling of the final product). Through an intelligent system of diagnostics and recommendations, VIDEXPERT provides users with customized guidelines to help reduce the carbon footprint.

The main objective of this thesis is to design and implement the data collection module of VIDEXPERT. This component requires an intuitive and friendly graphical user interface that facilitates users entering, validating and storing data on the processes, methods and products used during the wine production processes for subsequent analysis and calculation of the carbon footprint. The application of VIDEXPERT is expected to standardize the measurement of greenhouse gas emissions in the wine industry, facilitate the adoption of sustainable practices and optimize wine production processes.

During the development of VIDEXPERT, the agile Scrum methodology has been followed, using **Angular** as **Framework** for the frontend, **Spring Boot** for the backend and **MongoDB** as database.

As a future improvement, it is proposed to incorporate artificial intelligence to optimize the carbon footprint mitigation recommendations.

3. Agradecimientos

Quiero dedicar este trabajo a todas las personas que, de manera directa e indirecta, contribuyeron la finalización de mis estudios universitarios.

En primer lugar, mi gratitud a mi familia por el apoyo tanto emocional como económico. Su influencia ha sido clave para el desarrollo de mi carrera académica y han influenciado en mi forma de ver el mundo y de solucionar los problemas.

Un agradecimiento especial a mis tutores del Proyecto Fin de Grado (PFG), Miguel Ángel López Peña, Isabel Muñoz y a la profesora Pilar Manzano García. Su guía, dedicación y aportaciones han sido fundamentales para el desarrollo de este PFG. Gracias por compartir los conocimientos y la experiencia que me han ayudado a perfeccionar este PFG.

Agradezco a la Universidad Politécnica de Madrid (UPM) por la formación recibida y las oportunidades brindadas, con un reconocimiento especial a mis compañeros.

Por último, pero no menos importante, agradezco a SATEC por la oportunidad de formar parte de su equipo y por impulsar mi crecimiento profesional.

4. Introducción

En los últimos años, el cambio climático y la creciente demanda de productos sostenibles han impulsado a los productores de vino a reducir la huella de carbono en los procesos de vinificación. En España, donde este sector representa uno de los pilares fundamentales del comercio exterior, con un impacto socioeconómico y medioambiental de gran alcance, esta transición cobra especial relevancia. La normativa vigente refuerza este compromiso, exigiendo a las empresas el registro detallado de sus emisiones de CO₂ derivadas de actividades industriales. Como respuesta a esta necesidad, se ha desarrollado VIDEXPERT, una aplicación web diseñada para facilitar el cálculo de la huella de carbono y proporcionar recomendaciones inteligentes para su mitigación. A diferencia de otras opciones del mercado que no normalizan ni validan los datos de entrada (WINEKIA o Australian Wine Carbon Calculator (AWCC)), VIDEXPERT permite una gestión modular y escalable, lo que facilita futuras ampliaciones y mejoras.

VIDEXPERT es una aplicación web basada en una [arquitectura de microservicios](#) (posteriormente en la [Figura 1: Microservicios de la aplicación VIDEXPERT](#) se explicará con detalle), capaz de calcular las emisiones de CO₂ y ofrecer recomendaciones personalizadas para reducir el impacto ambiental. Esta herramienta inteligente diagnostica la huella de carbono mediante una serie de encuestas y preguntas que abarcan todas las actividades de las fases de la producción del vino, desde el cultivo de la uva hasta el embotellado del producto final. A partir de los datos recolectados, se calcula la huella de carbono del usuario y se generan recomendaciones personalizadas para su mitigación.

Este Trabajo de Fin de Grado (TFG) se centra en el módulo de recolección de datos de VIDEXPERT, que forma parte de dos microservicios principales: videxpert-core-client y videxpert-co2. Los sistemas actuales de recolección y cálculo de emisiones CO₂ se basan en formularios Excel, siendo inadecuados para la lectura y comprensión de los procedimientos y fórmulas. A esto se le suma la imposibilidad de validar los datos y la inexistencia de un estándar. El módulo de recolección de datos desarrollado en este TFG utiliza tecnologías modernas como **Angular**, **Spring Boot** y una base de datos no relacional **MongoDB**.

4.1. Objetivos del proyecto

Diseñar e implementar el módulo de recolección de datos de VIDEXPERT, que permita a los usuarios introducir, validar y almacenar información de manera eficiente para el cálculo de la huella de carbono en el sector vitivinícola. Para ello, se seguirán los siguientes pasos:

1. Desarrollar una interfaz gráfica intuitiva y amigable utilizando **Angular**.
2. Diseño de un modelo de datos que sea válido para mapear distintos formatos de datos.
3. Mapeo de las preguntas proporcionadas por los expertos al formato diseñado.
4. Implementar un sistema de control y validación de los datos para garantizar la calidad y consistencia de la información introducida.

5. Creación y almacenamiento de las respuestas recopiladas en un formato homogéneo para su posterior procesamiento.
6. Integrar el módulo de recolección de datos con los demás componentes de VIDEXPERT.

4.2. Estructura del documento

Este documento está estructurado de la siguiente manera:

- **Introducción:** descripción del contexto, justificación y objetivos.
- **Fundamentos teóricos y técnicos:** contenido sobre conceptos teóricos y técnicos que se necesita conocer para el entendimiento del proyecto.
- **Estado del arte:** análisis de tecnologías y herramientas existentes.
- **Aplicación VIDEXPERT:** descripción de la aplicación general, la metodología de trabajo y las herramientas utilizadas.
- **Módulo de recolección de datos (survey-section):**
 - **Análisis de requisitos:** descripción de requisitos funcionales y no funcionales.
 - **Modelado de datos:** descripción de tipos de datos utilizados.
 - **Diseño del módulo:** librerías, arquitectura, clases, interfaz de usuario y diagramas de flujo.
 - **Implementación:** descripción del desarrollo del código e integración con otros microservicios.
 - **Pruebas y validación:** resultados de las pruebas realizadas.
- **Discusión:**
 - Análisis de resultados obtenidos
 - Mejoras y limitaciones
 - Impacto.
- **Bibliografía y anexos:** referencias utilizadas y material adicional (código fuente, manuales).

5. Fundamentos teóricos y técnicos

En este apartado se definen los conceptos fundamentales y terminología técnica requeridos para mejorar el entendimiento del desarrollo de este proyecto. Se estructurará en tres puntos clave:

1. Conceptos teóricos: que aborda fundamentos ambientales y generales.
2. Información técnica: donde se describe herramientas y arquitecturas de implementación.
3. Librerías y lenguajes: que explica las librerías y los lenguajes que se han usado.

5.1. Conceptos teóricos

Emisiones de CO₂

Gases de efecto invernadero (GEI) [8] liberados por actividades humanas que contribuyen al calentamiento global. Los principales son dióxido de carbono (CO₂), metano (CH₄) y óxido nitroso (N₂O). Se miden en toneladas equivalentes de CO₂ para estandarizar su impacto ambiental. Nota: En este proyecto, "Emisiones de CO₂" incluye todos los gases de efecto invernadero relevantes.

Huella de carbono

Indicador que cuantifica el total de GEI emitidos directa o indirectamente por un producto, organización o individuo, considerando todo su ciclo de vida (producción, transporte, comercio y desecho). Este concepto es central para el proyecto, ya que su reducción es el objetivo principal.

Sector vitivinícola

Industria dedicada a la producción de vino [23], desde el cultivo de la vid hasta la comercialización del producto final. El proyecto se enfoca en este sector para reducir su impacto ambiental, sin profundizar en los detalles de sus procesos productivos.

MITECO

Ministerio para la Transición Ecológica y el Reto Demográfico de España [9]. Regula políticas ambientales, energéticas y climáticas, incluyendo normativas sobre la huella de carbono y sostenibilidad industrial. Su Registro Nacional de Huella de Carbono [10] es una certificación clave para el sector vitivinícola, cuya obtención se pretende facilitar mediante el desarrollo de este proyecto.

ISO

Organización Internacional de Estandarización que establece normas técnicas globales.

ODS

Objetivos de Desarrollo Sostenible [22]. 17 objetivos propuestos por la ONU que se espera cumplir para 2030 para erradicar la pobreza, proteger el planeta y garantizar prosperidad. La **ONU** (Organización de Naciones Unidas) ofrece marcos para reportar avances en sostenibilidad. Se explicará el impacto del proyecto relacionándolo con los ODS.

5.2. Información técnica

Librerías (TypeScript)

Código reutilizable que resuelve problemas específicos sin imponer una estructura global. Son vitales para evitar repetir código ya implementado.

Framework

Conjunto de herramientas y bibliotecas predefinidas que agilizan el desarrollo de software. Proporciona una estructura base para evitar código repetitivo.

Frontend

Capas de una aplicación con las que el usuario interactúa directamente. Se encarga de mostrar datos al usuario, gestionar interacción sistema-usuario usando las APIs del backend e incluye interfaces gráficas, animaciones y validación básica de datos. Se utilizará el framework **Angular** para el desarrollo de esta parte.

Interfaz web/gráfica

Parte visual e interactiva de una aplicación accesible mediante navegadores. Combina diseño (CSS), estructura (HTML) y funcionalidad (TypeScript) para permitir a los usuarios introducir datos, navegar y obtener resultados de forma intuitiva.

Backend

Capas de una aplicación que procesa la lógica de negocio, accede a bases de datos y garantiza seguridad. Hemos utilizado el lenguaje Java (**Spring Boot**), que se comunica con el frontend mediante APIs REST.

APIs REST

Representational State Transfer es un tipo de interfaz que permite la comunicación entre sistemas (cliente-servidor) usando el protocolo HTTP/HTTPS. Utiliza operaciones CRUD e intercambia información en formato JSON.

Operaciones CRUD

Conjunto de operaciones para manejo de datos:

- Create: crear (para HTTP es POST)
- Read: ver/leer (para HTTP es GET)
- Update: actualizar (para HTTP es PUT)
- Delete: eliminar

Arquitectura de microservicios

Enfoque arquitectónico que divide una aplicación en servicios pequeños, independientes y especializados. Cada microservicio tiene su propia base de datos y se comunica mediante APIs, mejorando escalabilidad y mantenimiento.

Stakeholders

Personajes clave afectados por un proyecto (clientes, equipos técnicos, inversores, reguladores...). Sus necesidades y opiniones guían el desarrollo del proyecto. Los desarrolladores no forman parte de los stakeholders.

Requisitos funcionales y no funcionales

Los requisitos son criterios que los stakeholders piden para cumplir con sus expectativas. [18]

- Funcionales: acciones concretas que el sistema tiene que ejecutar para satisfacer las necesidades de los usuarios.
- No funcionales: criterios o atributos de calidad. Pueden ser:
 - Escalabilidad: capacidad de un sistema para manejar crecimiento sin perder rendimiento.
 - Flexibilidad: adaptación a cambios con mínimo esfuerzo.
 - Mantenibilidad: facilidad para corregir errores o añadir funcionalidades.

Diagramas JSON

Representaciones gráficas de estructuras de datos en formato JSON (clave-valor). Se utiliza para modelar bases de datos NoSQL como **MongoDB**. Muestran jerarquías y tipos de datos.

Diagramas UML

Un diagrama UML [19] es una forma de visualizar sistemas y software utilizando el Lenguaje Unificado de Modelado (UML). Se utilizará para explicar los diseños, la arquitectura del código y la implementación de la aplicación. Algunos de los diagramas usados son:

- Clases: atributos, métodos y relaciones entre objetos
- Componentes: módulos del sistema y sus dependencias.
- Actividad: flujos de trabajo paso a paso.
- Casos de uso: interacciones entre usuarios y sistema.
 - Casos de uso extendido: versión detallada en formato de tablas. Cubre escenarios alternativos, condiciones de inicio.

URL

Uniform Resource Locator es una dirección única de un recurso en internet. Tienen un protocolo (http:// o https://), dominio, ruta y otros parámetros opcionales. En nuestra aplicación se utilizará <https://videxpert.benderlab.es/vid-expert> .

Pruebas unitarias

Validación automática de componentes individuales.

Pruebas de interfaz gráfica

Automatización de interacciones con la UI (interfaz de usuario, botones, llenado de formularios) para detectar errores visuales o de flujo.

Pruebas de aceptación

Evaluación final donde el cliente verifica que el sistema cumple con los requisitos acordados.

5.3. Librerías y lenguajes

5.3.1. Desarrollo de frontend

HTML

HTML (HyperText Markup Language) es el lenguaje estándar para crear páginas web. Define la estructura de las encuestas y las preguntas de la interfaz de usuario gracias a:

- Etiquetas: permite definir el significado del contenido.
- Integración con CSS y JavaScript.
- Fácil de aprender: lenguaje base para el desarrollo web.

Además, se utiliza Bootstrap [2], un **Framework** que mejora el diseño y la interactividad con la interfaz de usuario.

TypeScript

TypeScript es una ampliación de JavaScript, cuya diferencia principal es que el primero tiene tipado estático (se tiene que declarar el tipo de cada variable), y JavaScript tiene tipado dinámico. TypeScript es un lenguaje de programación de alto nivel orientado a objetos. Se utiliza para desarrollar la lógica de los componentes y servicios de **Angular**.

Sus características principales son:

- Interfaces interactivas: permite interactividad sin recargar la página porque se ejecuta en el navegador del usuario.
- Eventos: responde a acciones del usuario en tiempo real mediante eventos.
- Compatibilidad con todos los navegadores actuales.
- Asíncrono y no bloqueante: promesas y async/await permite manejar operaciones asíncronas.
- Entorno amplio: tiene gran cantidad de librerías y **Frameworks**.
- Tipado estático: controla errores limitando el tipo de una variable.
- Integración con HTML y CSS: manipulación del DOM (Document Object Model) para cambiar el contenido y estilo de una página web en tiempo real.

Angular

Angular [1] es un **Framework** mantenido por Google para desarrollo frontend basado en TypeScript. Es ampliamente utilizado para crear aplicaciones web dinámicas y escalables

como VIDEXPERT. Las principales razones para elegir Angular para el desarrollo del frontend son:

- División en componentes: permite dividir la interfaz en componentes reutilizables.
- Two-way data binding: sincronización automática entre la vista y el modelo.
- Inyección de dependencias: facilita la gestión de servicios y dependencias.
- CLI (Command Line Interface): herramienta que simplifica la creación, construcción y despliegue de proyectos.

Bootstrap

Bootstrap [2] es un **Framework** de diseño frontend que facilita la creación de interfaces web reactivas, adaptativas y modernas. Las características principales son:

- Sistema de cuadrículas: para crear plantillas de respuesta de manera sencilla.
- Componentes predefinidos, como botones, formularios, etc.
- Personalizable: adapta el diseño a las necesidades del proyecto.

5.3.2. Desarrollo de backend

Java

Java es un lenguaje de programación orientado a objetos. Se utiliza para desarrollar el backend con **Spring Boot**. Características principales:

- Multiplataforma: funciona en cualquier sistema operativo gracias a la JVM.
- Gran ecosistema: cuenta con una amplia variedad de librerías y **Frameworks**.
- Seguridad: proporciona herramientas para desarrollar aplicaciones seguras.
- Alto rendimiento: optimizado para aplicaciones de gran escala.

Spring Boot

Spring Boot [17] es un **Framework** de Java que simplifica el desarrollo de aplicaciones backend, especialmente microservicios y APIs RESTful. Se utiliza en el backend de VIDEXPERT para gestionar la comunicación con la base de datos MongoDB.

5.3.3. Bases de datos

MongoDB

MongoDB [4] es una base de datos NoSQL que almacena datos en formato de documentos JSON-like (BSON). Es altamente escalable y flexible, ideal para manejar datos no estructurados o semiestructurados. Guarda las preguntas, respuestas y resultados de las encuestas de VIDEXPERT en un formato homogéneo.

6. Estado del arte

6.1. Historia y necesidades

Actualmente, las empresas medianas y grandes en España están obligadas a calcular y registrar su **Huella de carbono**, y a elaborar un plan de reducción de **Emisiones de CO₂**. Para facilitar el proceso, el **MITECO** ofrece el Registro de Huella de Carbono [10], donde las empresas pueden inscribirse voluntariamente y registrar sus cálculos y planes de reducción de emisiones. El MITECO emite además otros certificados, entre ellos, destaca Sustainable Wineries for Climate Protection (SWfCP) [12] en el sector vitivinícola. Esta certificación se centra en la sostenibilidad ambiental de las bodegas, evaluando aspectos como eficiencia energética, gestión del agua, reducción de emisiones y gestión de residuos. Gracias a dicha certificación, las empresas del sector vitivinícola demuestran su compromiso con la sostenibilidad y mejoran su competitividad en el mercado. Además, facilita el acceso al Registro Nacional de Huella de Carbono del MITECO y otorga una mayor puntuación en los Programas de Apoyo al Sector Vitivinícola.

6.2. Soluciones existentes

En el sector vitivinícola en Europa, la medición precisa de la **Huella de carbono** es esencial para implementar prácticas sostenibles y obtener certificaciones del **MITECO**. Sin embargo, no existe estándar para el cálculo de la huella de carbono, y la obtención de datos confiables y precisos resulta una tarea compleja. Esto supone una oportunidad de negocio que han visto muchas organizaciones. Algunas herramientas son:

- **Winekia** [15] es una plataforma en línea que permite a las bodegas calcular la huella de carbono de sus productos. Ya está disponible en línea y facilita el proceso de obtención de algunos certificados [16]. A diferencia de VIDEXPERT, Winekia está diseñado solo para etapas específicas (viñedo y bodega), no ofrece sugerencias de mitigación.
- **Australian Wine Carbon Calculator (AWCC)** es una aplicación desarrollada por asociaciones vitivinícolas australianas. Esta herramienta está basada en formularios Excel, lo que dificulta el análisis y mantenimiento de grandes volúmenes de datos.
- **WineGB Farm Carbon Calculator** es un sistema web para productores de Inglaterra y Gales. Tiene mayor accesibilidad que soluciones basadas en Excel, pero no muestra sus fuentes de datos o cómo se hacen los cálculos.
- **California Code of Sustainable Winegrowing Workbook** es una herramienta basada en EXCEL y con limitaciones de usabilidad y escalabilidad.

A raíz de las limitaciones, se ha desarrollado **VIDEXPERT**. Esta es una herramienta inteligente que calcula y diagnostica la huella de carbono, y ofrece recomendaciones personalizadas para mitigar el impacto ambiental durante todas las etapas de producción del vino. Nuestro Proyecto Fin de Grado se basa en esta aplicación. Ya se ha completado su fase de desarrollo y pronto estará desplegada [14].

7. VIDEXPERT

VIDEXPERT [13] es una aplicación web desarrollada por un consorcio de entidades, entre las que se encuentran INTERGIA, SATEC y la Federación Española del Vino (FEV) [11]. La aplicación está basada en una arquitectura de **microservicios**, lo que permite una mayor escalabilidad, flexibilidad y mantenibilidad. Los principales microservicios que componen VIDEXPERT son los siguientes:

- **videxpert-core-client**: se encarga de mostrar la interfaz de usuario (UI) e implementa todas las funcionalidades necesarias para la navegación por la aplicación, así como la recolección, estructuración y validación de datos. Utiliza HTML, TypeScript (TS) y SCSS para el diseño y la interactividad.
- **videxpert-co2**: actúa como intermediario entre la base de datos (donde se almacenan las encuestas, preguntas y respuestas) y la interfaz de usuario. Implementa los endpoints y la lógica necesaria para transmitir los datos de manera eficiente. Utiliza Java y **Spring Boot**.
- **co2-calculator**: calcula la **Huella de carbono** en base a los datos recopilados por **videxpert-co2**. Este módulo es clave para proporcionar métricas precisas sobre las **Emisiones de CO₂**.
- **recommendation-service**: genera recomendaciones personalizadas para la reducción de la huella de carbono, basándose en los cálculos proporcionados por **co2-calculator**. Este servicio es fundamental para guiar a los usuarios en la implementación de prácticas más sostenibles.
- **geographic service**: realiza el cálculo de distancias utilizando un servicio externo. En futuras versiones, se planea integrar este módulo con el servicio SIGPAC (Sistema de Información Geográfica de Parcelas Agrícolas), lo que permitirá una geolocalización más precisa de las parcelas en España.
- **videxpert-core**: gestiona la creación e identificación de los usuarios de la herramienta, garantizando un acceso seguro y personalizado a la aplicación.

En la [Figura 1](#) se puede ver una imagen visual de los microservicios descritos anteriormente.

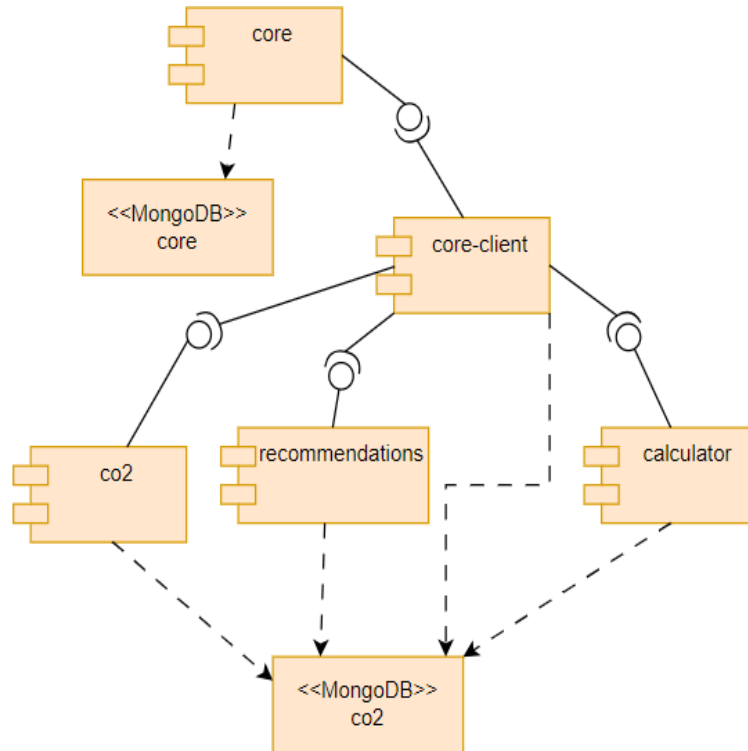


Figura 1: Microservicios de la aplicación VIDEXPERT

VIDEXPERT maneja un volumen diverso de tipos de datos (número, texto, zonas geográficas, datos temporales, selección única) provenientes de las múltiples etapas de la producción del vino. Esta diversidad de información hace que el módulo de recolección de datos, integrado en los microservicios videxpert-co2 (backend) y videxpert-core-client (frontend), sea un componente esencial del sistema. Este módulo está diseñado para garantizar la captura eficiente y estructurada de la información para el cálculo de la huella de carbono y resolver las limitaciones de los métodos tradicionales de recolección de datos (hojas Excel, documentos en Word), como son la validación, la escalabilidad o el análisis.

Las ventajas clave son:

- Facilidad de obtención de certificaciones, reduciendo los márgenes de error por fallo humano y creando una base para la estandarización del cálculo en la huella de carbono en el sector vitivinícola.
- Mejora de la toma de decisiones gracias a datos fiables y consistentes.
- Análisis comparativos entre diferentes bodegas
- Trazabilidad de los cambios, manteniendo un registro completo y automatizado.

Este Trabajo de Fin de Grado (TFG) profundiza en la implementación técnica de este módulo fundamental, demostrando cómo su arquitectura innovadora resuelve los desafíos actuales en la recolección y gestión de datos ambientales para la industria del vino.

7.1. Metodología de trabajo

Este apartado explica el desarrollo del módulo de recolección de datos mediante encuestas en el proyecto VIDEXPERT de la empresa SATEC. Durante la gestión del proyecto, se ha seguido una metodología ágil [5] (Scrum) [6] con reuniones de seguimiento diarias.

Se ha decidido utilizar una metodología ágil debido a su naturaleza iterativa e incremental, donde los requisitos del proyecto evolucionan y/o cambian con el tiempo y es necesario entregar funcionalidades de manera progresiva. Esto permite fácilmente afrontar los cambios que pueden surgir durante las diferentes etapas del desarrollo software. Características del enfoque ágil:

Colaboración continua: el equipo de desarrollo trabaja en estrecha colaboración con los **Stakeholders**.

- Iteraciones (Sprints): de 2 a 4 semanas, donde se planifican y desarrollan distintas funcionalidades del sistema. Al final de cada Sprint se entrega una versión funcional del producto.
- Adaptabilidad: mediante los cambios en los requisitos del proyecto durante las distintas etapas del desarrollo software.
- Entregas incrementales: Al final de cada sprint, se ha seguido una metodología ágil (Scrum), con reuniones de seguimiento diarias.

Scrum

Se ha seguido la metodología Scrum [7] durante el desarrollo del proyecto de VIDEXPERT. Esto permite gestionar el trabajo de manera iterativa e incremental. En detalle, se ha seguido el flujo de trabajo de la [Figura 2](#):

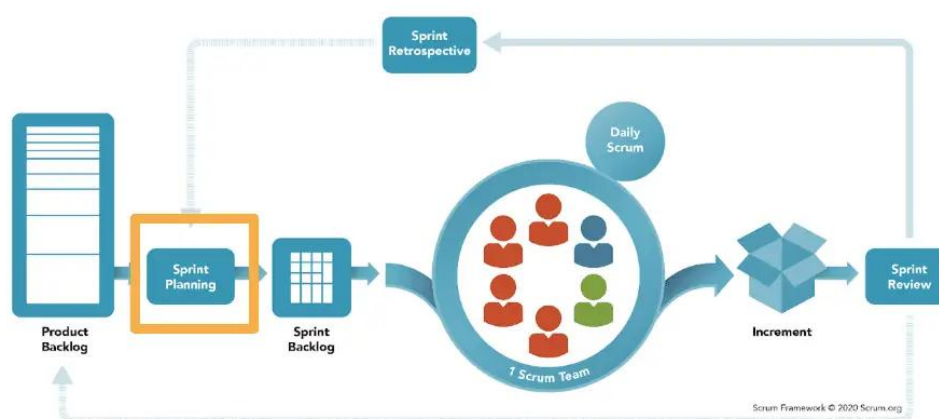


Figura 2: Ciclo del Scrum [21]

1. Descomposición: divide el problema en sus partes más pequeñas que se listan en el Product Backlog
2. Sprint Planning: reunión de planificación donde se selecciona un conjunto de tareas a ejecutar durante el Sprint, que se listan en el Sprint Backlog.

3. Develop: desarrollo de las tareas seleccionadas desde el Sprint Backlog. Durante el desarrollo del Sprint, se mantiene reuniones diarias llamadas Daily Scrum donde se comparten avances e inconvenientes.
4. Sprint Review: reunión de revisión con los **Stakeholders** para obtener retroalimentación de comentarios.
5. Sprint Retrospective: reunión de retrospectiva para analizar lo que funcionó bien, lo que se puede mejorar y las lecciones aprendidas durante el sprint. En la [Figura 3](#) se puede ver un ejemplo de las tareas del Product Backlog en la aplicación de JIRA.

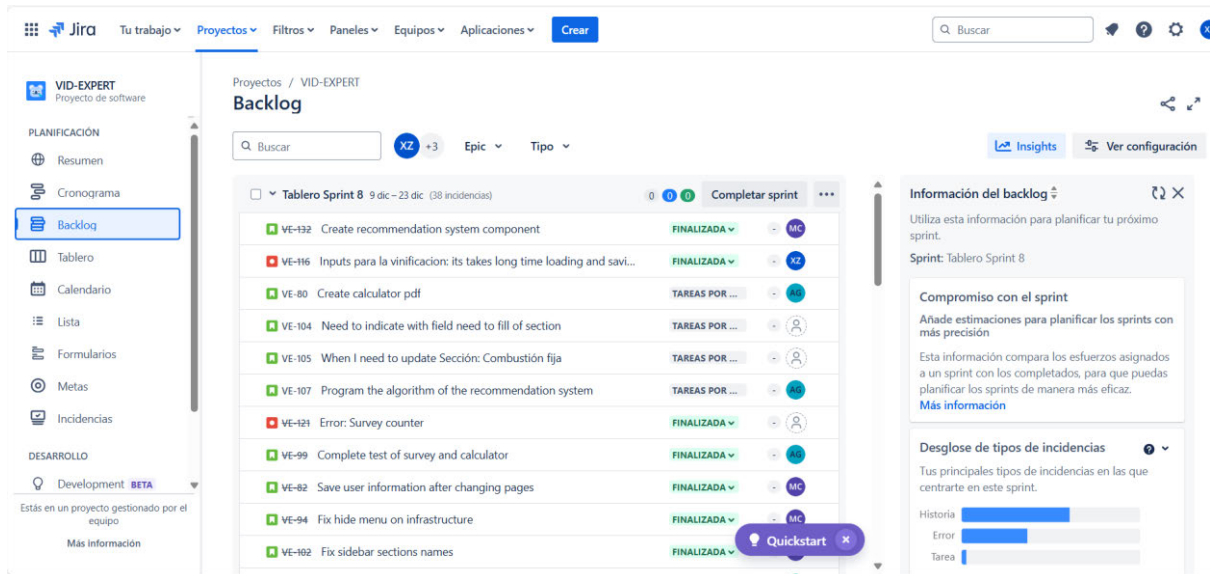


Figura 3: Product backlog del Jira

7.2. Aplicaciones externas de apoyo

En este apartado se detallarán las herramientas y aplicaciones externas que se han sido requeridas para el desarrollo de VIDEXPERT.

IntelliJ IDEA

IntelliJ IDEA [3] es un IDE (Entorno de Desarrollo Integrado) desarrollado por JetBrains. Se utiliza como el entorno principal para desarrollar tanto el frontend (**Angular**[1]) como el backend (Spring Boot) de VIDEXPERT. Permite:

- Soporte multiplataforma: disponible para Windows, macOS y Linux.
- Integración con **Frameworks**: soporta **Spring Boot** y **Angular**.
- Herramientas de refactorización: facilita la mejora y mantenimiento del código.
- Depurador avanzado: permite identificar y corregir errores de manera eficiente.

MongoDB Compass

MongoDB Compass [4] es una interfaz gráfica para MongoDB que permite visualizar, consultar y gestionar datos de manera intuitiva. Se utiliza para gestionar y visualizar los datos en local para garantizar que los datos se hayan guardado correctamente. Características principales:

- Exploración de datos: permite navegar por las colecciones y documentos.
- Consultas visuales: facilita la creación de consultas.
- Análisis de esquemas: Muestra la estructura de los documentos en la base de datos.
- Integración con MongoDB Atlas: Soporta bases de datos en la nube.

Git y GitHub

Es un sistema de control de versiones y plataforma de colaboración para el desarrollo del código. Permite a distintos programadores trabajar sobre un mismo proyecto evitando conflictos de código.

Jira

Jira es una herramienta de gestión de proyectos diseñada para equipos que utilizan metodologías ágiles, como Scrum. Esta herramienta se ha usado principalmente para gestionar las tareas dentro del equipo de desarrollo de VIDEXPERT, permitiendo una fácil organización y priorización de las tareas y el progreso.

8. Requisitos del módulo

En este apartado se explicarán los requisitos que debe tener el módulo de recolección de datos para satisfacer las necesidades de los clientes. El módulo de recolección de datos es una parte vital de VIDEXPERT, encargada de formatear, validar y recolectar la información. Se explicarán los requisitos funcionales y no funcionales [18] y se utilizará un diagrama de casos de uso.

8.1. Requisitos funcionales

- RF-01: El sistema deberá permitir a los usuarios crear nuevas encuestas para calcular las **Emisiones de CO₂** en un determinado año.
- RF-02: El sistema deberá permitir a los usuarios modificar el nombre y la descripción de sus propias encuestas.
- RF-03: El sistema deberá permitir a los usuarios visualizar las respuestas de una encuesta.
- RF-04: Los usuarios de rol "ADMIN" podrán visualizar todas las encuestas de todos los usuarios.
- RF-05: El sistema deberá permitir a los usuarios eliminar sus propias encuestas, con previa confirmación.
- RF-06: El sistema deberá permitir a los usuarios navegar por las distintas secciones (mediante breadcrumb o sidebar o navbar).
- RF-07: El sistema deberá permitir a los usuarios elegir completar una de entre varias subsecciones en las secciones que sean tipo "subsection_election".
- RF-08: El sistema deberá permitir a los usuarios añadir una fila de respuestas para las subsecciones de respuesta múltiple.
- RF-09: El sistema deberá permitir a los usuarios eliminar filas de respuestas para las subsecciones de respuesta múltiple.
- RF-10: El sistema deberá permitir a los usuarios completar y modificar las respuestas.
- RF-11: El sistema validará automáticamente la entrada de los datos en tiempo real, advirtiendo de datos requeridos, formatos o tipo de entrada.
- RF-12: El sistema permitirá que el usuario guarde las respuestas de una subsección con previa comprobación de la validez de los datos.
- RF-13: El sistema deberá identificar las subsecciones y secciones completadas por los usuarios de una encuesta y mostrarlo en tiempo real.
- RF-14: El sistema deberá permitir a los usuarios hacer el cálculo de la huella de carbono cuando todas las secciones estén completas.
- RF-15: El sistema tendrá un tutorial interactivo de cómo funciona la aplicación.

8.2. Requisitos no funcionales

- RNF-01 (Rendimiento): La aplicación debe responder a las solicitudes del usuario en menos de 30 segundos.
- RNF-02 (Integridad): Los usuarios solo pueden modificar las respuestas de sus propias encuestas.

- RNF-03 (Integridad): Los usuarios solo pueden eliminar sus propias encuestas.
- RNF-04 (Usabilidad): La interfaz debe ser intuitiva, con un tiempo de aprendizaje menor a 10 minutos para nuevos usuarios.
- RNF-05 (Compatibilidad): La interfaz web debe ser compatible con Chrome, Firefox, Edge y Safari (últimas 2 versiones).

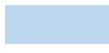





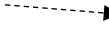

9. Modelo de datos

En esta sección se explican la estructura de datos del módulo de recolección de VIDEXPERT. Se utilizan los tipos básicos de TypeScript como base, pero también existen estructuras más complejas especialmente diseñadas para el diseño del sistema. Los tipos básicos son:

- string: datos textuales, cadenas de caracteres
- number: números enteros y decimales
- boolean: valor lógico verdadero/falso (true/false)
- any: tipo genérico que permite datos dinámicos
- Array<T>: lista ordenada de elementos tipo T. Se representan como T [] (ej. string[] o number[])
- dateTime: fecha en formato ISO¹ string

Las estructuras complejas de los datos del módulo de VIDEXPERT está compuesto por 6 tablas principalmente: Section, Subsection, Item, Question, Answer y Survey. Cada elemento de la base de datos tiene un ID creado automáticamente mediante **MongoDB**.

Para explicar la estructura de los datos, se ha usado varios diagramas de JSON representados en la [Figura 4](#). Hay que marcar que la nomenclatura usada ha sido la siguiente:

-  Recuadro azul: nombre de la tabla.
-  Recuadro amarillo: atributo compartido con la base de datos.
-  Recuadro verde: atributo propio del módulo.
-  Recuadro rosa: atributos de relación embebida.
-  Recuadro naranja: valores constantes de los nombres de atributos.
-  Flechas continuas: relación directa entre atributos/componentes.
-  Flechas discontinuas: relación referenciada.
-  Líneas discontinuas: posibles valores del atributo.

¹ ISO: (International Organization for Standardization) formato más utilizado para representar fechas y horas en sistemas digitales

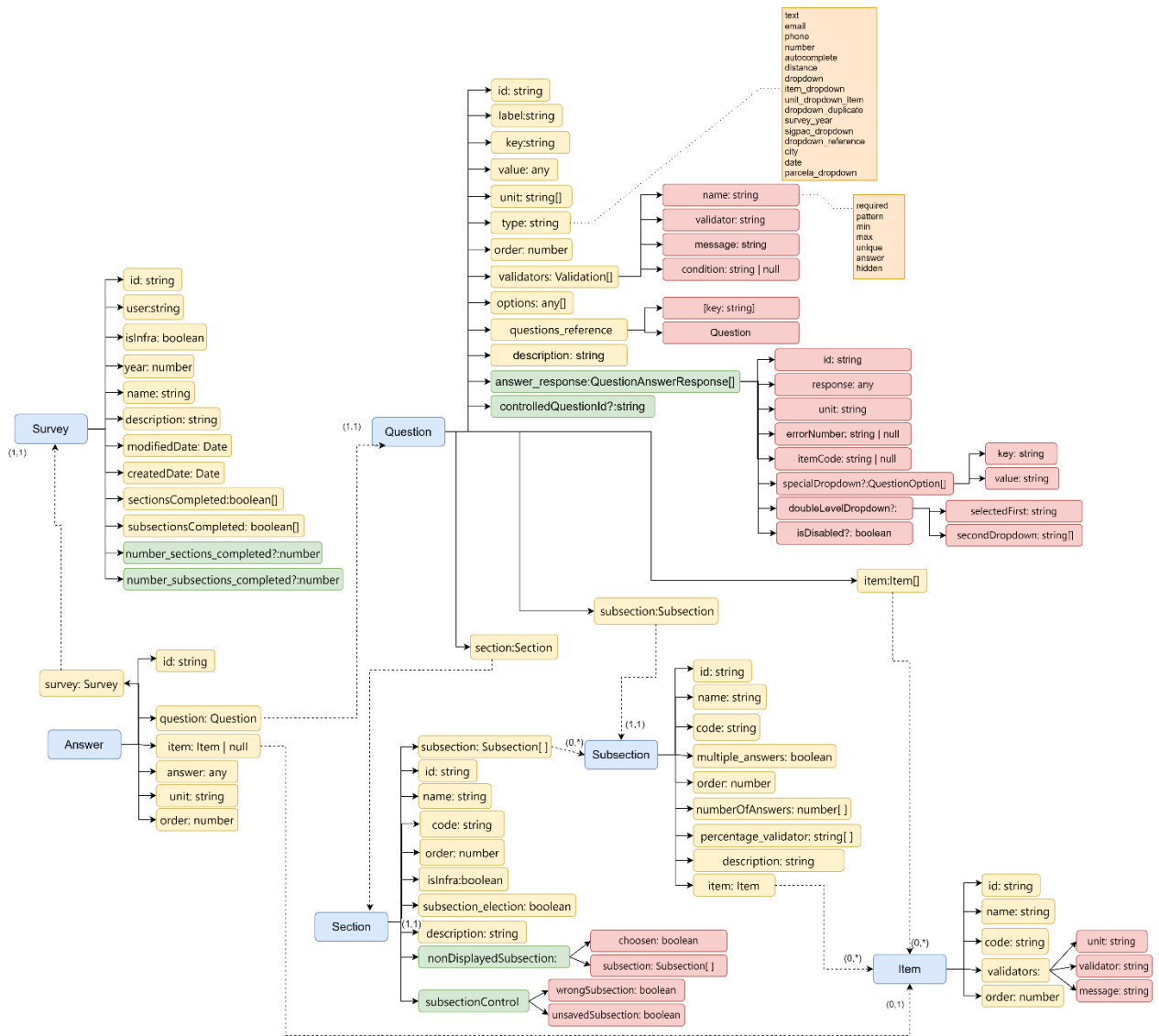


Figura 4: Tablas de estructuras de datos

Survey es la tabla donde se almacenan las encuestas de los usuarios. El cálculo de CO₂ se basa en una encuesta. Los principales atributos son:

- **userId (string)**: es el identificador del usuario que ha creado la encuesta. Debido a que la tabla de datos de usuarios y encuestas se encuentran en bases de datos distintas, el userId es un texto en plano y no una referencia. Al tener separado, se protege la privacidad del usuario, dificultando la consulta del propietario de cada encuesta.
- **isInfra (boolean)**: valor booleano que representa:
 - **true**: la encuesta es de tipo cálculo. Las respuestas de este tipo de encuestas se utilizarán para el cálculo de la huella de carbono. Cada usuario puede tener varias encuestas de este tipo por año de estudio.
 - **false**: la encuesta es de tipo infraestructura. Solo tiene valor informativo y complementario a las de tipo cálculo. Un usuario solo puede tener una encuesta de este tipo.

- year (number): año en el que se quiere realizar el estudio. Con este atributo se da más flexibilidad a la hora de crear encuestas de años anteriores y tener una trazabilidad de las encuestas realizadas.
- name (string): nombre de la encuesta.
- description (string): descripción de la encuesta.
- createdAt (Date): fecha de creación de la encuesta.
- modifiedAt (Date): última fecha de modificación.
- sectionsCompleted (boolean[]): lista de valores booleanos donde el "true" representa las secciones completas en ese orden.
- subsectionsCompleted (boolean[]): lista de valores booleanos donde el "true" representa las subsecciones completas en ese orden.
- number_sections_completed (number): cantidad de elementos 'true' en sectionsCompleted. Sirve para facilitar la visualización de secciones completas con respecto al total de secciones.
- number_subsections_completed (number): cantidad de elementos 'true' en subsectionsCompleted.

Section representa las secciones de las encuestas. Cada sección abarca un área de estudio. Los atributos que tiene son:

- name (string): nombre de la sección que se muestra al usuario.
- order (number): número de orden de la sección para organización.
- code (string): código único que identifica la sección. Se usa en [URLs](#) de navegación web y para cálculos (a diferencia del id).
- isInfra (boolean):
 - true: pertenece a encuestas de tipo "infraestructura".
 - false: pertenece a encuestas de tipo "cálculo" estándar.
- description (string): descripción detallada de la sección.
- subsection_election (boolean):
 - true: Permite al usuario elegir entre subsecciones para completar.
 - false: No permite elección (subsecciones fijas).
- nonDisplayedSubsections (object[]): solo si subsection_election=true, lista las subsecciones agrupadas por orden para elección.
 - chosen (boolean): indica si la subsección en este orden es elegible.
 - subsections (array): lista de subsecciones disponibles en orden.
- subsectionControl (object[]): lista de atributos por subsección que ayuda a controlar el flujo de datos en el frontend.
 - wrongSubsection: verdadero si la subsección tiene errores.
 - unsavedSubsection: verdadero si subsección tiene respuestas sin guardar. Además, si la subsección no tiene respuestas, también es verdadero.
- subsection (Subsection[]): lista de subsecciones (referencia a la tabla Subsection).

Subsection muestra subsecciones de la sección. Divide las áreas de estudio en unidades más específicas.

- name (string): nombre de la subsección que se muestra al usuario.
- order (number): número de orden para organización.
- code (string): código único que identifica la subsección.
- multiple_answers (boolean):
 - true: permite respuestas múltiples.
 - false: no permite respuestas múltiples.
- numberOfAnswers (number[]): lista ordenada del número de respuestas de la subsección.
- percentage_validator (string[]): lista de códigos (code) de preguntas (questions) cuya suma de respuestas debe ser 100%.
- description (string): descripción detallada de la subsección.
- item (Item[]): lista de items de la subsección.

Item campo de estudio con el que se puede asociar una pregunta.

- name (string): nombre que se muestra al usuario.
- order (number): número de orden para organización.
- code (string): código único que identifica el item.
- validators: requisitos de validación de las preguntas del item.
 - unit (string): filtro de las preguntas con dicha unidad.
 - validator (string): valor con el que se validan las preguntas.
 - message (string): mensaje de error en caso de que no se cumpla el valor.

Question son las preguntas que hay que responder.

- section (Section): referencia única a la sección a la que pertenece.
- subsection (Subsection): referencia única a la subsección a la que pertenece.
- item (Item[]): lista de items (Item) que tiene la pregunta.
- label (string): nombre de la pregunta que se muestra al usuario.
- key (string): clave única y estática para hacer referencias.
- value (any): valor por defecto de la pregunta.
- unit (string[]): lista de unidades que se le permiten a las respuestas (Answer)
- type (string): tipo de pregunta. Pueden ser:
 - text: texto.
 - email: email. Debe tener el formato de email (texto sin espacio y contiene un @ en medio)
 - phone: número de teléfono de 9 dígitos
 - number: número.
 - autocomplete: el valor por defecto depende de otra pregunta y se obtiene a partir de una operación en value.
 - distance: calcula por defecto la distancia entre dos preguntas de tipo city. También puede ser editado manualmente.
 - city: doble lista desplegable que muestra la lista de países de Europa y las ciudades de cada una. Las listas se obtienen de una librería externa de **Angular** llamada .npmjs [20].

- parcela_dropdown: doble lista desplegable que muestran la lista de parcelas y recintos.
- dropdown: lista desplegable con respuesta fija en options.
- item_dropdown: en options tiene una lista desplegable a dos niveles. El primer nivel depende de la question.key que aparece en questions_reference. El segundo nivel es la lista de desplegables que se mostrará al usuario en formato QuestionOption² que depende de la respuesta a la que referencia.
- unit_dropdown_item: en options tiene una lista desplegable a tres niveles. Este tipo de preguntas depende de las respuestas de dos preguntas (questions_reference.unit y questions_reference.dropdown). El tercer nivel es la lista de desplegables que se mostrará al usuario en formato QuestionOption.
- dropdown_duplicate: lista desplegable a dos niveles. El primer nivel depende de la question.key que aparece en questions_reference. El segundo nivel es una lista de desplegables en formato QuestionOption que depende de la respuesta a la que referencia. A diferencia de item_dropdown, este tipo muestra todas las opciones del segundo nivel.
- survey_year: lista desplegable que muestra los meses de octubre del año anterior al de la encuesta hasta diciembre del mismo año de la encuesta.
- sigpac_dropdown: utiliza el microservicio de sigpac_service para ir mostrando la lista de respuestas siguiendo el orden de comunidad, provincia, municipio, polígono, parcela, recinto.
- dropdown_reference: lista desplegable que utiliza las respuestas de la pregunta a la que referencia en questions_reference.dropdown como opciones.
- date: las respuestas tienen formato DateTime.
- order (number): orden en el que se muestran las preguntas.
- validators (Validation[]): validan si la pregunta es válida.
 - name (string): nombre del validator. Puede ser
 - required: la respuesta no puede ser vacía.
 - pattern: la respuesta tiene que seguir el patrón de validators.
 - min: la respuesta tiene un valor mínimo.
 - max: la respuesta tiene un valor máximo.
 - hidden: deshabilita la respuesta si se cumple el validators.
 - answer: la pregunta a la que se asocia el key de condition tiene que cumplir con el valor de validator.
 - validator (string): requisito del validators.
 - message (string): mensaje de error cuando no se cumple el validator.
 - condition (string | null): si no se cumple la condition, el validator no se aplica. En caso contrario o cuando no hay condition, el validator se aplica.

² QuestionOption{
 key: string,
 value: string
 }

- options (any[]): lista de opciones para las listas desplegables.
- questions_reference ({{key:string}:Question}): lista de preguntas a las que hace referencia.
- description (string): texto descriptivo de la pregunta.
- answer_response (QuestionAnswerResponse[]): respuesta temporal guardada en el frontend permitiendo validación en tiempo real y reduciendo el número de llamadas al backend.
 - response (any): respuesta temporal del usuario. Cada vez que éste cambia, se valida.
 - unit (string): unidad de la respuesta.
 - errorNumber (string | null): mensaje de error en caso de respuesta inválida.
 - itemCode (string | null): código del item al que hace la referencia. Tiene que ser un item válido, es decir, estar en la lista de item del atributo de la tabla Question.
 - specialDropdown? (QuestionOption[]): en algunos tipos de preguntas, se necesita hacer una transformación del atributo option para adecuarlo a una lista de desplegables siguiendo el formato:
 - key (string): valor que se guarda como respuesta en la base de datos.
 - value (string): valor mostrado en la lista de desplegables.
 - countryDropdown? ({{ selectedCountry: string}, {countryList: string[]}): lista de desplegables especial para las preguntas tipo city. Tiene dos desplegables, uno lista los países y otro las ciudades. Al seleccionar un país, el segundo desplegable se actualiza con las ciudades de dicho país. Este método reduce los fallos al insertar los nombres de las ciudades manualmente.
 - isDisabled? (boolean): atributo que habilita o deshabilita la respuesta de dicha pregunta.
- controlledQuestionId? (string): id de la pregunta a la que hace referencia. Sirve para encontrar dinámicamente la pregunta sin necesidad de volver a llamar al servicio para recuperar las preguntas de la base de datos.

Para entender mejor la lista de desplegables, en la [Figura 5](#) se ha hecho un esquema explicativo de type y options.

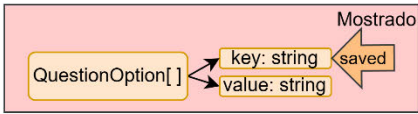
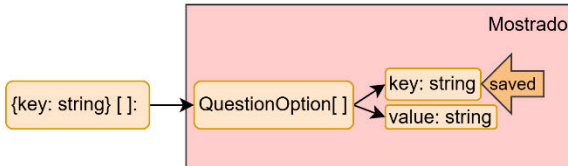
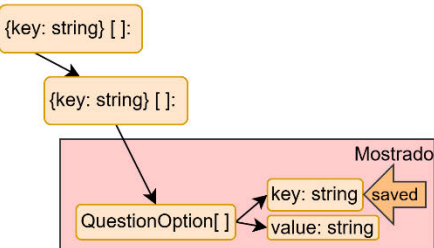
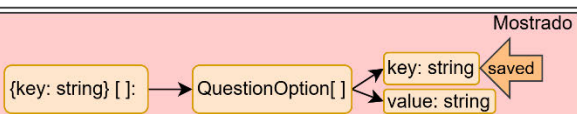
type	option
dropdown	 <pre> "options": [{ "key": "Aerothermal", "value": "Bomba de calor aerotérmica" }, { "key": "Geothermal", "value": "Bomba de calor geotérmica" }], </pre>
item_dropdown	 <pre> "options": [{ "car": [{ "key": "COM-GAS-01", "value": "Gasolina" }], "van": [] }], </pre>
unit_dropdown_item	 <pre> "options": [{ "ud. combustible": [{ "car": [{ "key": "COM-E5-01", "value": "Gasolina/E5 (litros)" }], "km": [] }] }], </pre>
dropdown_duplicate	 <pre> "options": [{ "van": [{ "key": "van", "value": "Furgoneta" }], "motorcycle": [] }], </pre>

Figura 5: Tipos de desplegables

Se ha usado la notación “Mostrado” para definir los valores que se muestran al usuario; y “saved” para indicar que es el valor guardado como respuesta en *Answer*.

En la [Figura 6](#) se muestra un ejemplo de la interfaz de usuario de las preguntas tipo city, survey_year y date.

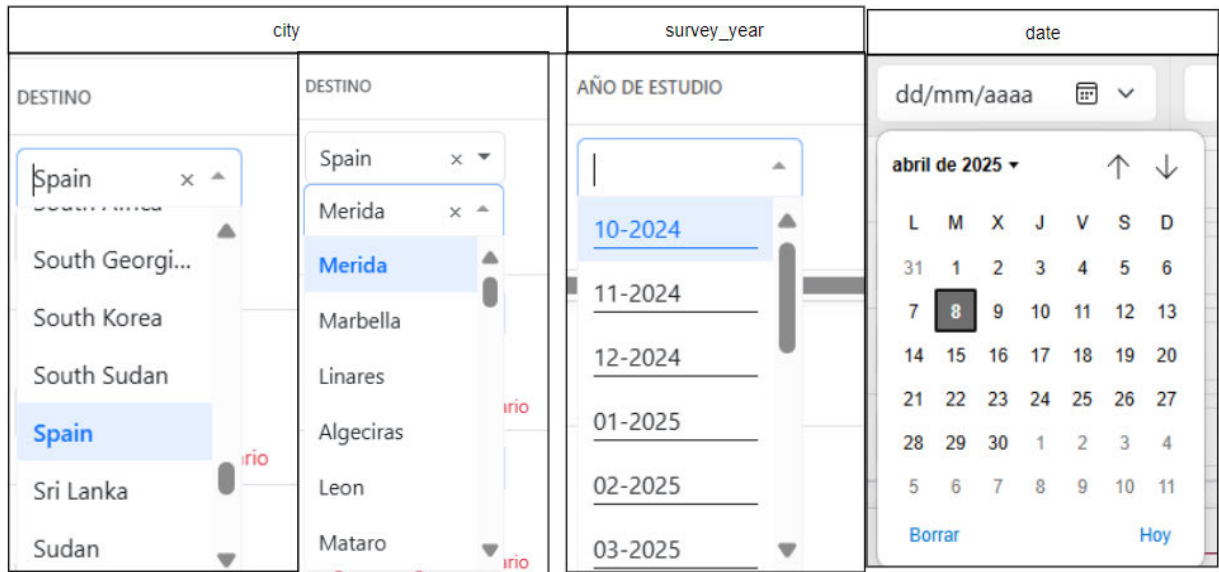


Figura 6: Ejemplo de city, survey_year, date

Answers es la tabla con las respuestas de los usuarios

- question (Question): referencia a la tabla Question.
- survey (Survey): referencia a la tabla Survey.
- item (Item | null): posible referencia a la tabla Item.
- answer (any): respuesta del usuario. Puede ser un string, number, phone, email, date o string[]. Las preguntas que son desplegables solo guardan el valor de key.
- unit (string): unidad de medida de la respuesta.
- order (number): orden de la respuesta. Sirve para ordenar las respuestas para las preguntas de respuesta múltiple.

9.1. Ejemplos visuales de datos

En este apartado se mostrarán ejemplos visuales de cómo varía la interfaz de usuario en las subsecciones y secciones.

La Figura 7 es un ejemplo visual de Subsección de respuesta múltiple. Esta subsección permite añadir o eliminar filas y guardar toda la subsección.

2 : CARACTERÍSTICAS GENERALES DE LAS PARCELAS ⓘ

Guardar subsección +

Mostrar 5 elementos << < 1 2 3 4 > >>

ÚMERO DE	VARIEDAD	AÑO DE PLANTACIÓN (PROMEDIO)	ANCHURA DE LA CALLE (PROMEDIO)	DISTANCIA ENTRE PLANTAS	SUMA DEL NÚMERO DE FILAS DE LOS RECINTOS	SISTEMA DE CONDUCCIÓN
	Sauvignon Blanc	2021,29	3 m	1,5 m	11	Espaldera
	Cabernet sauvignon	2006	3 m	1,5 m	43	Espaldera

Figura 7: Subsección de respuesta múltiple

La **Figura 8** muestra una sección con ítem. Las preguntas están en la primera fila, mientras que los ítem están en la primera columna. Las preguntas que no tienen algunos ítem vienen en la celda sombreada.

1: GESTIÓN DE RESIDUOS, REVALORIZACIÓN Y RECICLAJE EN BODEGA ⓘ Guardar subsección

PREGUNTA:	CANTIDAD GENERADA	PORCENTAJE DESTINADO A DESECHAR (BASURA GENERAL)	PORCENTAJE DESTINADO A SEPARACIÓN PARA RECICLADO	PORCENTAJE DESTINADO A REÚSO	PORCENTAJE DESTINADO A REVALORIZACIÓN	PORCENTAJE DESTINADO A VENTA A TERCEROS
1. Hollejos	<input type="text" value="0"/> kg	<input type="text" value="0"/> %		<input type="text" value="0"/> %	<input type="text" value="0"/> %	<input type="text" value="0"/> %
2. Lías	<input type="text" value="0"/> kg	<input type="text" value="0"/> %		<input type="text" value="0"/> %	<input type="text" value="0"/> %	<input type="text" value="0"/> %
3. Papel y cartón	<input type="text" value="0"/> kg	<input type="text" value="0"/> %	<input type="text" value="0"/> %	<input type="text" value="0"/> %	<input type="text" value="0"/> %	<input type="text" value="0"/> %
4. Plásticos	<input type="text" value="0"/> kg	<input type="text" value="0"/> %	<input type="text" value="0"/> %	<input type="text" value="0"/> %		<input type="text" value="0"/> %
5. Vidrio	<input type="text" value="0"/> kg	<input type="text" value="0"/> %	<input type="text" value="0"/> %	<input type="text" value="0"/> %		<input type="text" value="0"/> %
6. Restos	<input type="text" value="0"/> kg	<input type="text" value="0"/> %		<input type="text" value="0"/> %	<input type="text" value="0"/> %	<input type="text" value="0"/> %

Figura 8: Subsección con ítem, pero algunas preguntas no tienen ítem

En la **Figura 9** se muestra una subsección con preguntas normales que vienen en forma de lista, mientras que las preguntas con ítem vienen en tabla.

1: SUMINISTRO DE LA ELECTRICIDAD DE LA RED ⓘ Guardar subsección

CUPS

Porcentaje de uso de la energía para Riego viñedos %

Porcentaje de uso de Otras aplicaciones en viñedo %

Porcentaje de uso de la energía para Consumo en bodega %

Porcentaje de uso de la energía para Consumo en espacios enoturismo %

PREGUNTA:	CONSUMO MENSUAL	NOMBRE DE LA COMERCIALIZADORA	GDO	PORCENTAJE RENOVABLE
1. Noviembre	<input type="text" value="number"/> kWh	<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value="number"/> %
2. Diciembre	<input type="text" value="number"/> kWh	<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value="number"/> %
3. Enero	<input type="text" value="number"/> kWh	<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value="number"/> %
4. Febrero	<input type="text" value="number"/> kWh	<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value="number"/> %
5. Marzo	<input type="text" value="number"/> kWh	<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value="number"/> %
6. Abril	<input type="text" value="number"/> kWh	<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value="number"/> %

Figura 9: Subsección con preguntas normales e ítems

En la [Figura 10](#) se muestran dos subsecciones, una con `subsection_election`(recuadro de abajo) y otra que no tiene (recuadro de arriba). Cada subsección viene separada y recuadrada.

1: CONSUMO DE AGUA EN BODEGA ⓘ Guardar subsección

Cantidad de agua de red empleada por la bodega m3 ▾

Cantidad de agua vertida a la red sin depurar m3 ▾

Elegir completar una de las subsecciones: Consumo de agua de riego total en la organización ▾

2: CONSUMO DE AGUA DE RIEGO TOTAL EN LA ORGANIZACIÓN ⓘ Guardar subsección

Cantidad agua m3 ▾

Figura 10: Sección con `subsection_election` y la otra no

10. Diseño del módulo

El módulo de recolección de datos está estructurado en componentes **Angular**. Se ha diseñado una interfaz de usuario intuitiva en HTML utilizando Bootstrap, que permite a los usuarios introducir datos de manera eficiente. Los datos se validan antes de ser almacenados en una base de datos NoSQL, asegurando su coherencia y calidad. Se utilizarán diagramas **UML** para facilitar la explicación.

10.1. Diagrama de casos de uso

El diagrama de casos de uso (Figura 11) muestra las acciones que un tipo de usuario puede realizar sobre el sistema. Con este diagrama se explican las funciones que se le permiten al usuario.

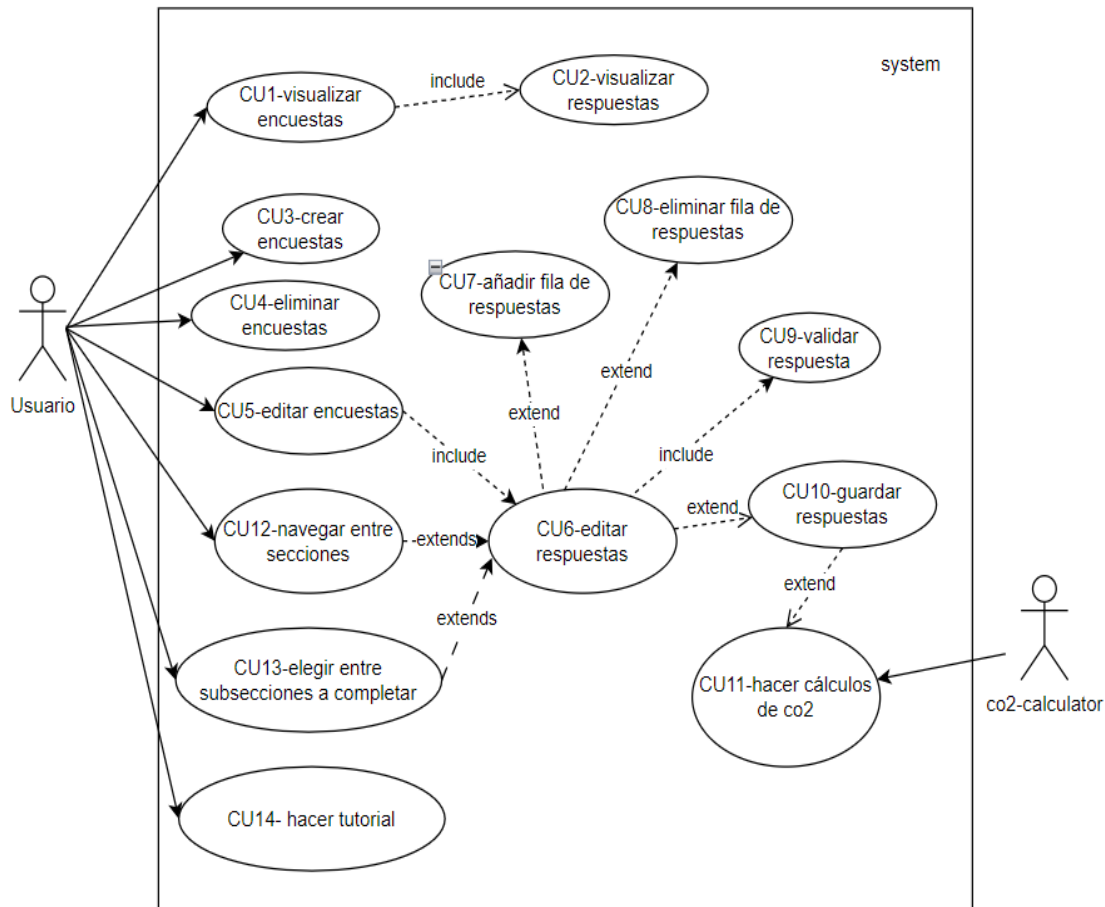


Figura 11: Diagrama de casos de uso

A continuación, mediante las tablas de casos de uso extendido se explicará en detalle cada caso de uso. La notación usada será la que se indica en la tabla siguiente:

Identificador	Código y nombre identificativo del caso de uso. Está asociado con el que aparece en la Figura 11 .
Requisito asociado	Requisitos funcionales que abarca dicho caso de uso. Está asociado con los requisitos descritos en 8.1 Requisitos funcionales .
Descripción	Breve descripción de las acciones que puede hacer el usuario.
Precondición	Condiciones que se tienen que cumplir para permitir empezar el caso de uso.
Flujo normal	Conjunto de pasos que sigue el sistema en caso de que la acción se haya iniciado correctamente.
Flujo alternativo	Paso en el que podría ocurrir una interrupción del flujo normal, y acciones que toma el sistema en dicha interrupción para finalizar correctamente el caso de uso.

Teniendo en cuenta la nomenclatura anterior, se explicarán en orden los casos de uso de la [Figura 11](#).

Identificador	CU1-Visualizar encuesta
Requisito asociado	RF-03: El sistema deberá permitir a los usuarios visualizar las respuestas de una encuesta. RF-04: Los usuarios de rol "ADMIN" podrán visualizar todas las encuestas de todos los usuarios.
Descripción	El usuario podrá visualizar los detalles de la encuesta.
Precondición	El usuario tiene que ser un usuario válido de la lista de usuarios.
Flujo normal	1. El usuario pulsa sobre un cuestionario para ver sus detalles
Flujo alternativo	<i>No aplica</i>

Identificador	CU2-Visualizar respuestas
Requisito asociado	RF-03: El sistema deberá permitir a los usuarios visualizar las respuestas de una encuesta. RF-04: Los usuarios de rol "ADMIN" podrán visualizar todas las encuestas de todos los usuarios.
Descripción	El usuario podrá visualizar las respuestas de las encuestas ya completadas.
Precondición	El usuario tiene que ser un usuario válido de la lista de usuarios. Debe seleccionar la encuesta de la que quiere ver la respuesta.
Flujo normal	1. El usuario pulsa "Ver respuestas" de un cuestionario.
Flujo alternativo	1. Si no existe el cuestionario, avisa al usuario y finaliza.

Identificador	CU3-Crear encuesta
Requisito asociado	RF-01: El sistema deberá permitir a los usuarios crear nuevas encuestas para calcular las Emisiones de CO₂ en un determinado año.
Descripción	El usuario podrá crear nuevas encuestas rellenando el nombre, detalle y año de la encuesta.
Precondición	El usuario tiene que ser un usuario válido de la lista de usuarios.
Flujo normal	1. El usuario crea una encuesta.
	2. El usuario rellena el nombre, la descripción y el año.
	3. El sistema crea la encuesta y la guarda en la base de datos.
Flujo alternativo	2. Hay errores en algún campo de la encuesta. Vuelve al paso 2 del flujo normal.

Identificador	CU4-Eliminar encuesta
Requisito asociado	RF-05: El sistema deberá permitir a los usuarios eliminar sus propias encuestas, con previa confirmación.
Descripción	El usuario podrá eliminar una encuesta que sea suya.
Precondición	El usuario tiene que ser un usuario válido de la lista de usuarios.
Flujo normal	1. El usuario selecciona una encuesta
	2. El usuario pulsa sobre eliminar encuesta
	3. Se pide confirmación para eliminar la encuesta con todas las respuestas de dicha encuesta
	4. El usuario acepta borrar la encuesta
	5. El sistema borra la encuesta con las respuestas asociadas.
Flujo alternativo	4. El usuario cancela. Finaliza.

Identificador	CU5-Editar encuesta
Requisito asociado	RF-02: El sistema deberá permitir a los usuarios modificar el nombre y la descripción de sus propias encuestas
Descripción	El usuario podrá editar los atributos de la encuesta.
Precondición	El usuario tiene que ser un usuario válido de la lista de usuarios. Debe existir la encuesta
Flujo normal	1. El usuario selecciona una encuesta
	2. El usuario modifica un atributo de la encuesta
	3. El usuario guarda los cambios
Flujo alternativo	3. El usuario cancela los cambios. Finaliza.

Identificador	CU6-Editar respuestas
Requisito asociado	RF-07: El sistema deberá permitir a los usuarios elegir completar una de entre varias subsecciones en las secciones que sean tipo "subseccion_election". RF-10: El sistema deberá permitir a los usuarios completar y modificar las respuestas
Descripción	El usuario puede editar las respuestas de una encuesta.
Precondición	El usuario tiene que ser un usuario válido de la lista de usuarios. La encuesta tiene que existir en la base de datos.
Flujo normal	1. El usuario modifica una respuesta.
	2. El sistema valida la entrada de datos (CU9)
	3. El usuario guarda los cambios de la subsección (CU10).
Flujo alternativo	1. El usuario elige una subsección a completar. Ir al paso 1 del flujo normal.
	1. El usuario añade una fila de respuestas. Ir a CU7.
	1. El usuario elimina una fila de respuestas. Ir a CU8.
	3. El usuario cancela los cambios y no guarda. Finaliza

Identificador	CU7-Añadir fila de respuesta
Requisito asociado	RF-08: El sistema deberá permitir a los usuarios añadir una fila de respuestas para las subsecciones de respuesta múltiple.
Descripción	El usuario deberá poder añadir filas de respuestas a las subsecciones de respuesta múltiple.
Precondición	El usuario tiene que ser un usuario válido de la lista de usuarios. La subsección tiene que ser de respuesta múltiple.
Flujo normal	1. El usuario añade una fila de respuestas.
	2. El usuario puede editar las respuestas (CU6).
Flujo alternativo	<i>No aplica</i>

Identificador	CU8-Eliminar fila de respuesta
Requisito asociado	RF-09: El sistema deberá permitir a los usuarios eliminar filas de respuestas para las subsecciones de respuesta múltiple.
Descripción	El usuario deberá poder eliminar filas de respuestas de las subsecciones de respuesta múltiple.
Precondición	El usuario tiene que ser un usuario válido de la lista de usuarios. La subsección tiene que ser de respuesta múltiple. La subsección debe tener al menos una fila de respuestas.
Flujo normal	1. El usuario elimina una fila de respuestas.
	2. El sistema pide confirmación al usuario.
	3. El usuario acepta eliminar la fila de respuestas.
Flujo alternativo	3. El usuario cancela la acción. Finaliza.

Identificador	C9-Validar respuesta
Requisito asociado	RF-11: El sistema validará automáticamente la entrada de los datos en tiempo real, advirtiendo de datos requeridos, formatos o tipo de entrada.
Descripción	El sistema deberá validar automáticamente las respuestas modificadas.
Precondición	El usuario tiene que ser un usuario válido de la lista de usuarios. El usuario debe haber editado una respuesta.
Flujo normal	1. El sistema valida si el formato de la respuesta corresponde con el tipo de pregunta.
	2. El sistema valida a nivel de pregunta.
	3. El sistema valida a nivel de ítem.
	4. El sistema valida a nivel de subsección.
	5. Se da por correcta la respuesta.
Flujo alternativo	1. Falla el tipo de pregunta y salta mensaje de error. Finaliza.
	2. Falla a nivel de pregunta y salta mensaje de error. Finaliza
	3. Falla a nivel de ítem y salta mensaje de error. Finaliza
	4. Falla a nivel de subsección y salta mensaje de error. Finaliza.

Identificador	C10-Guardar respuesta
Requisito asociado	RF-12: El sistema permitirá que el usuario guarde las respuestas de una subsección con previa comprobación de la validez de los datos.
Descripción	El sistema guarda las respuestas por subsección con previa comprobación
Precondición	El usuario tiene que ser un usuario válido de la lista de usuarios. El usuario debe haber editado las preguntas.
Flujo normal	1. El sistema comprueba la validez de los datos.
	2. El sistema guarda los datos en la base de datos.
	3. Salta mensaje de haber guardado las respuestas correctamente.
Flujo alternativo	2. Hay error en alguna respuesta y salta mensaje de error. Finaliza.

Identificador	CU11-Hacer cálculos de co2
Requisito asociado	RF-14: El sistema deberá permitir a los usuarios hacer el cálculo de la huella de carbono cuando todas las secciones estén completas.
Descripción	El usuario deberá poder hacer el cálculo de la huella de carbono de la encuesta una vez haya completado todas las secciones.
Precondición	El usuario tiene que ser un usuario válido de la lista de usuarios. La encuesta debe estar completa.
Flujo normal	1. El usuario pulsa el botón para hacer la huella de carbono.
	2. El sistema llama a un servicio externo para hacer los cálculos.
	3. Se muestra un mensaje de cálculo hecho con éxito.
Flujo alternativo	2. Falla el sistema externo y se muestra un mensaje de error. Finaliza.

Identificador	CU12-Navegar entre secciones
Requisito asociado	RF-13: El sistema deberá identificar las subsecciones y secciones completadas por los usuarios de una encuesta y mostrarlo en tiempo real. RF-06: El sistema deberá permitir a los usuarios navegar por las distintas secciones (mediante breadcrumb o sidebar o navbar).
Descripción	El usuario puede navegar entre las distintas secciones de la encuesta y mostrar su progreso.
Precondición	El usuario tiene que ser un usuario válido de la lista de usuarios. Debe existir la encuesta.
Flujo normal	1. El usuario selecciona una sección a la que navegar. 2. El sistema comprueba si hay cambios sin guardar. 3. El sistema carga los datos de dicha subsección.
Flujo alternativo	2. Se muestra mensaje de cambios sin guardar. 2a. El usuario guarda los cambios. 2a1. El sistema guarda los cambios de las respuestas. Ir a 3 del flujo normal. 2b. El usuario no quiere guardar los cambios. Ir a 3 del flujo normal. 2c. El usuario cancela la acción de cambiar de sección. Finaliza.

Identificador	CU13-elegir entre subsecciones a completar
Requisito asociado	RF-07: El sistema deberá permitir a los usuarios elegir completar una de entre varias subsecciones en las secciones que sean tipo "subsection_election".
Descripción	El usuario puede elegir completar una de las subsecciones.
Precondición	El usuario tiene que ser un usuario válido de la lista de usuarios. La sección debe tener "subsection_election" a true.
Flujo normal	1. El usuario selecciona una sección que quiere completar.
Flujo alternativo	<i>No aplica</i>

Identificador	CU14-Hacer tutorial
Requisito asociado	RF-15: El sistema tendrá un tutorial interactivo de cómo funciona la aplicación.
Descripción	El usuario puede hacer un tutorial interactivo sobre las funciones de la aplicación.
Precondición	<i>No aplica</i>
Flujo normal	1. El usuario selecciona hacer el tutorial. 2. El sistema muestra el tutorial paso a paso.
Flujo alternativo	<i>No aplica</i>

10.2. Librerías externas

En esta sección se detallan las dependencias externas y módulos de **Angular** implementados en el sistema, fundamentales para su arquitectura y funcionalidad.

- RouterModule: Librería fundamental para el sistema de navegación entre páginas (SPA - Single Page Application). Permite definir rutas, redirecciones y parámetros de **URL**.
- NgbModule (Ng Bootstrap): Proporciona componentes UI de Bootstrap adaptados para Angular (modales, tooltips, dropdowns, etc.).
- FormsModule: Soporte para formularios template-driven (con ngModel).

- **ReactiveFormsModule**: Permite crear formularios complejos dinámicos o con lógica avanzada.
- **xlsx (SheetJS)**: Librería JavaScript de código abierto diseñada para leer, manipular y escribir archivos de hojas de cálculo en formatos como Excel (XLSX, XLS, CSV).
- **FeatherIconModule**: Integra los iconos de Feather Icons en la aplicación.
- **NgbModalModule**: Módulo específico de Ng Bootstrap para ventanas NgbModal.
- **sweetalert2**: Librería moderna para reemplazar los diálogos (alert, confirm, prompt) con ventanas modales altamente personalizables y responsivas.
- **NgSelectModule**: Componente personalizable para dropdowns/selects con búsqueda (ng-select).
- **HttpClientModule**: Módulo para hacer peticiones HTTP (GET, POST, etc.) a APIs REST.
- **NgxMaskModule**: Librería para enmascarar inputs (formatos de teléfono, email).
- **TranslateModule**: Parte de ngx-translate, para cambiar idiomas dinámicamente sin recargar la aplicación.
- **BehaviorSubject**: Clase especializada de la librería RxJS (Reactive Extensions for JavaScript) que actúa como un tipo de Observable que almacena el último valor emitido ("estado actual") y lo envía inmediatamente a nuevos suscriptores. Con esta clase, se puede compartir datos entre componentes/clases no relacionados.
- **driver.js**: Librería dedicada a hacer tutoriales iterativos y personalizables.
- **city-timezone**: lista de países y ciudades de Europa. [\[20\]](#)

10.3. Diagrama de clases

El diagrama de clases describe la estructura y relaciones de las clases del componente. Se explicarán en detalle los componentes section ([Figura 12](#)) y survey ([Figura 13](#)) debido a que son más complejos y los que más peso tienen en este Proyecto Fin de Grado.

El componente section se encarga de:

1. Mostrar lista de secciones, subsecciones y preguntas.
2. Gestionar rutas y redirección dinámica: dirige al usuario a las distintas páginas según el tipo de encuesta y el usuario. El atributo dinámico sectionCode muestra la sección actual al usuario.
 - `/infrastructure/sectionCode`: encuestas de infraestructura (datos de contacto y datos físicos de bodegas/viñedos: maquinaria, construcciones, parcelas...). Se completa una sola vez por usuario.
 - `/survey/sections/surveyId/sectionCode`: encuestas de tipo cálculo: Con ellas se calcula las **Emisiones de CO₂** por año de estudio. Se basan en las actividades y las gestiones realizadas por los viñedos y las bodegas. Requiere datos previos de la encuesta de infraestructura. El surveyId es el id de la encuesta actual que el usuario está completando.
 - Redirigir automáticamente al usuario en caso de no ser autor de la encuesta a:
 - o `/viewInfrastructure/surveyId/sectionCode`: vista de solo lectura para encuestas de tipo infraestructura.

- /viewSections/surveyId/sectionCode: vista de solo lectura para encuestas de tipo cálculo.
3. Validar datos de entrada del usuario antes de enviarlos al backend.
 4. Controlar el progreso en tiempo real del número de secciones/subsecciones completadas.
 5. Tutorial guiado para ayudar a los usuarios a completar las secciones correctamente.

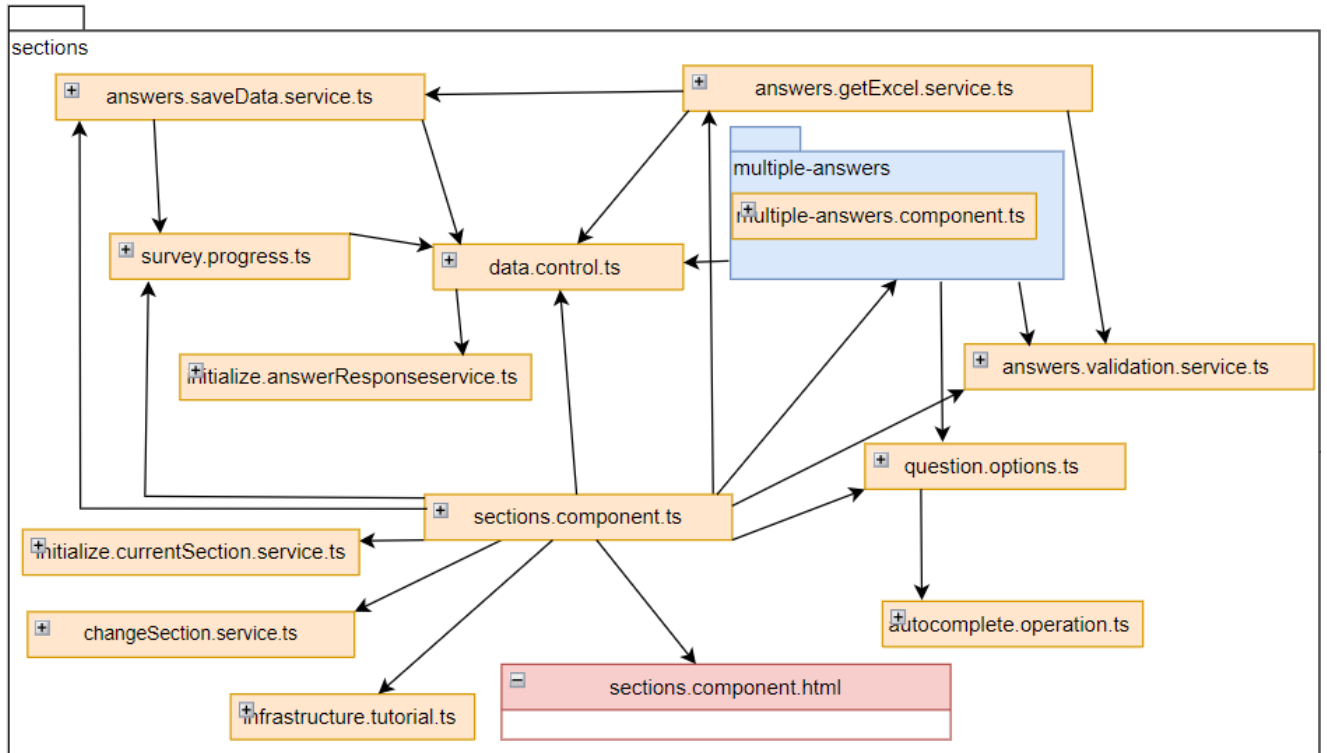


Figura 12: Diagrama de clases (sections)

En las siguientes tablas se explica el funcionamiento de cada clase.

+ indica que el atributo/función/servicio es público (cualquier clase puede acceder a ella).

- indica que el atributo/función/servicio es privado (solo los métodos de dicha clase pueden acceder a ella)

* indica que el atributo/función /servicio es protegido (solo la propia clase y sus subclases/herederas pueden acceder a ella).

Nombre	sections.component.ts
Descripción	Esta clase es el componente principal del módulo. Controla directamente el funcionamiento de la interfaz de usuario mostrado en el HTML. Se conecta al resto de servicios del sistema.
Atributos	<pre> +survey: Survey +user: User +sections: Section[] = [] +questionsOfSection: Question[] = [] +currentSection: Section +answersFromSection: Answer[] = [] +unsavedAnswers: boolean[] = [] +isSavingSubsection: boolean +isSurveyCompleted: boolean = false +isCalculatorUpdated: boolean = true -deleteConfirmationModal: NgbModalRef undefined -surveyDetailsModal: NgbModalRef undefined </pre>
Métodos	<pre> +ngOnInit(): void {} - checkSectionHasWrongAnswers(questionsOfSection: Question[]): void -initializeFirstSectionByCode(sectionList: Section[]):void +getQuestionsFromSection(section: Section):void +chooseDisplayedSubsection(section: Section, answersFromSection: Answer[]): Section +filterQuestionsBySubsection(subsect: Subsection): Question[] +filterNoItemQuestionByItemSubsection(subsect: Subsection): Question[] +filterItemQuestionByItemSubsection(subsect: Subsection): Question[] +saveSubsection(subsection: Subsection): void +changeBreakcrumb(section: Section):void +changeSection(section: Section):void +changeLink(link: string) -changeToSectionLink(event: any, toSection: boolean): void -changeSectionOrRouterLink(toSection: boolean, destiny: any) +changeDisplayedSubsections(subsection: Subsection, order: number): void +changeControlledQuestion(question: Question, i: number): void +changeCountrySelection(question: Question, qAnswerResponse: QuestionAnswerResponse, i: number): void +addCustomCity(name: string): QuestionOption +isAnswerValid(question: Question, response: any, i: number): void +sortItem(elements: any[]): any[] +addAnswers(subsection: Subsection): void -checkSubsectionIsValid(subsection: Subsection):void +openDeleteConfirmationModal(content: TemplateRef<any>, subsection: Subsection, order: number): void +openSubsectionDetails(subsection: Subsection): void +openQuestionDetails(event: MouseEvent, question: Question): void +openSurveyDetailsModal(content: TemplateRef<any>): void +getSubsectionsOrderInSections(subsection: Subsection, section: Section): number +makeCalculations(): void +importExcelAnswers(event: any, subsection: Subsection): void +startTutorial() +ngOnDestroy() </pre>
Servicios (privados) utilizados	<pre> router: Router route: ActivatedRoute modalService: NgbModal, getCityService: GetCityService answerSavingService: AnswersSaveDataService surveyProgress: SurveyProgress, calculatorService: CalculatorService answerValidation: AnswersValidationService surveyDataControl: DataControl multipleAnswersComponent: MultipleAnswersService questionOptions: QuestionOptions </pre>

	infrastructureTutorial: InfrastructureTutorial, answersGetExcel: AnswersGetExcelService, translate: TranslateService
--	--

Nombre	sections.component.html
Descripción	Documento HTML que estructura la base de la interfaz de usuario.

Nombre	data.control.ts
Descripción	<p>Esta clase centraliza la lógica de negocio relacionada con encuestas para evitar código duplicado y mantener consistencia en la UI.</p> <p>Utiliza el patrón reactivo de Angular (RxJS) para propagar los cambios a todos los componentes interesados, es decir, usa observables (BehaviorSubject) que notifican automáticamente a los componentes suscritos cuando los datos cambian. Esto permite compartir datos reactivamente entre componentes (survey, sections, questions, answers), y coordinando los servicios de obtención de datos del backend para que el módulo siempre disponga de datos actualizados.</p> <p>Además, controla la navegación basada en permisos (redirigiendo a modo "solo lectura" si el usuario no es dueño de la encuesta).</p>
Atributos	<pre>+ user: User - survey: Survey - surveySubject = new BehaviorSubject<Survey null>(null) + survey\$ = this.surveySubject.asObservable() - sectionsSubject = new BehaviorSubject<Section[]>([]) + sections\$ = this.sectionsSubject.asObservable() - currentSectionSubject = new BehaviorSubject<Section null>(null) + currentSection\$ = this.currentSectionSubject.asObservable() - questionsSubject = new BehaviorSubject<Question[]>([]) + questionsOfSection\$ = this.questionsSubject.asObservable() - initialQuestionsSubject = new BehaviorSubject<Question[]>([]) + initialQuestions\$ = this.initialQuestionsSubject.asObservable() - answersFromSection: Answer[] - answersSubject = new BehaviorSubject<Answer[]>([]) + answersOfSection\$ = this.answersSubject.asObservable() - route: ActivatedRoute - urlLink: string</pre>
Métodos	<pre>+setSurvey(survey: Survey):void +setSections(section: Section[]):void +setCurrentSection(section: Section):void +setQuestions(question: Question[]):void +setInitialQuestion(question: Question[]):void +setAnswers(answers: Answer[]):void +getSurveyData(router: Router, route: ActivatedRoute):void -getUrlLink():void -getAllSections():void -getAllInfra():void -manageSection(sections: Section[], survey: Survey):void +getQuestionsFromSection(section: Section):void -getAnswersOfSection(questionsOfSection: Question[]): void restoreValues()</pre>
Servicios (privados) utilizados	<pre>userDataService: UserDataService surveyService: SurveyService sectionService: SectionService, questionService: QuestionService answerService: AnswerService, questionInitialize: DataInitialize, router: Router</pre>

Nombre	infrastructure.tutorial.ts
Descripción	Componente que se encarga de mostrar el tutorial para el usuario. Los pasos del tutorial están definidos en la variable infrastructureSteps, que utiliza driver.js como librería externa para simplificar la lógica y el diseño.
Atributos	+ driverObj - infrastructureSteps
Métodos	+startInfrastructureTutorial():void -updateDriverButtons():void
Servicios (privados) utilizados	- translate: TranslateService - driver: Driver.js

Nombre	initialize.answerResponse.ts
Descripción	Este servicio complementa a data.control.ts normalizando los datos para el frontend, Convierte la relación entre las entidades Answer y Question (originalmente en tablas separadas) en un formato embebido (QuestionAnswerResponse), donde cada pregunta incluye sus respuestas asociadas.
Atributos	
Métodos	+setControlledQuestionID(questions: Question[]): Question[] +setAnswerResponseOfQuestion(question: Question, answers: Answer[]):Question -setAnswerResponseNoItem(question: Question, answers: Answer[]):QuestionAnswerResponse[] -setAnswerResponseWithItem(subsect: Subsection, question: Question, answers: Answer[]): QuestionAnswerResponse[] -setAnswerResponse(response: Answer undefined,question: Question, qltem: Item undefined): QuestionAnswerResponse
Servicios (privados) utilizados	<i>No aplica</i>

Nombre	answers.saveData.service.ts
Descripción	Mapea (transforma QuestionAnswerResponse a Answer) y guarda las respuestas. Hay que destacar que es importante que las secciones de subsection_election eliminen las respuestas de la subsección que no ha sido elegida. Una vez guardadas las respuestas, el sistema llamará al servicio de actualizar el progreso de la encuesta.
Atributos	- survey: Survey - currentSection: Section - answersToSave: Answer[] - answersFromSection: Answer[] - questionsFromSection: Question[] - getAnswers: boolean
Métodos	+saveAnswers(currentSection: any): Promise<any> +saveAndGetAnswers(currentSection: Section, subsection: Subsection): Promise<any> -saveAnswersFromSubsection(currentSection: Section, subsection: Subsection): void -deleteAnswersFromHiddenSubsection(section: Section, subsectionOrder: number): void -saveAnswersFromQuestion(question: Question, answersFromQuestion: Answer[]): void -sentAnswerResponse(answRespo: QuestionAnswerResponse, order: number, question: Question): void -deleteAnswer(answerId: string): void -getAnswerResponseOfNewAnswer(answerDTO: AnswerDTO, question: Question): void
Servicios (privados) utilizados	answerService: AnswerService surveyProgress: SurveyProgress private surveyDataControl: DataControl translate: TranslateService

Nombre	survey.progress.ts
Descripción	Este servicio se encarga de capturar y actualizar el progreso de la encuesta (survey). Una sección se da por completada cuando todas sus subsecciones están completadas. Una subsección se da por completada cuando existe al menos una pregunta guardada o es una subsección de respuesta múltiple.
Atributos	-sections: Section[] -survey: Survey -questionsOfSection: Question[]
Métodos	+setSectionsCompleted(currentSection: Section, currentSubsection: Subsection): void +getSubsectionsOrderInSurvey(subsection: Subsection, section: Section): number
Servicios (privados) utilizados	surveyService: SurveyService surveyDataControl: DataControl

Nombre	answers.getExcel.service.ts
Descripción	Este servicio se encarga de cargar, validar y guardar las respuestas cargadas de un Excel. Se ha decidido permitir a los usuarios cargar los datos de las parcelas y recintos desde una tabla Excel para facilitar la migración de los datos desde el método tradicional al de usar la herramienta. Este servicio funciona de forma secuencial, siguiendo estos pasos: <ol style="list-style-type: none"> 1. Verificar el formato del formulario Excel se corresponde con el que se les proporciona. 2. Cargar los datos de dicho formulario. 3. Verifica que las respuestas siguen las reglas (validations) proporcionadas. 4. Guardar las respuestas en la Base de Datos.
Atributos	- enclosureExcelData: any[] - parcelExcelData: any[] - excelSubsectionsName: {enclosureSections: string, parcelasSections: string,} - questionsOfSection: Question[]; - currentSection: Section
Métodos	+setSectionsCompleted(currentSection: Section, currentSubsection: Subsection): void +getSubsectionsOrderInSurvey(subsection: Subsection, section: Section): number
Servicios (privados) utilizados	surveyService: SurveyService surveyDataControl: DataControl

Nombre	answers.validation.service.ts
Descripción	Este servicio valida las respuestas siguiendo el orden de question>item>subsection y ayuda en la inicialización de las preguntas al validar el valor por defecto de las preguntas. Si una respuesta es errónea, se verá reflejado el mensaje de error en QuestionAnswerResponse[i].errorNumber.
Atributos	survey: Survey; questionsOfSection: Question[];
Métodos	+checkQuestionValidators(question: Question, i: number, response: QuestionAnswerResponse): void +validateQuestion(response: any, question: Question, order: number):QuestionAnswerResponse -validateQuestionTypeFormat(questionType: string, response: any): string null -checkValidatorsCondition(validator: Validation, order: number, currentQuestion: Question): QuestionAnswerResponse -checkHiddenAnswerValidator(currentQuestion: Question, order: number): QuestionAnswerResponse -checkQuestionValidator(response: any, question: Question): string null -checkItemValidator(item: Item, order: number):void -checkSubsectionPercentageValidator(question: Question, order: number):void
Servicios (privados) utilizados	surveyDataControl: DataControl translate: TranslateService

Nombre	autocomplete.operation.ts
Descripción	Este servicio se encarga de dar un valor por defecto a las preguntas de tipo "autocomplete". Este tipo de preguntas utilizan la respuesta de otras preguntas y convierten, mediante una ecuación, en el valor por defecto de la pregunta para mostrar al usuario.
Atributos	<i>No aplica</i>
Métodos	evaluateExpression(expression: string, questionsOfSection: Question[], order: number): Promise<any>
Servicios (privados) utilizados	<i>No aplica</i>

Nombre	multiple-answers.component.ts
Descripción	Esta clase se encarga de todas las operaciones relacionadas con las subsecciones de respuesta múltiple: <ul style="list-style-type: none"> • Añadir una fila de respuestas. Se tiene que validar el valor por defecto de las preguntas. • Eliminar cualquier fila de respuestas, con previa confirmación del usuario. Los datos eliminados solo son en local y, hasta que el usuario no guarde la subsección, no se eliminarán de la base de datos. • Actualizar el número de respuestas de la subsección.
Atributos	- answersFromSection: Answer[]; - questionsOfSection: Question[];
Métodos	+getNumberOfAnswers(subsection: Subsection): number[] +addAnswers(subsection: Subsection): void +deleteLocalAnswers(subsection: Subsection, order: number): void
Servicios (privados) utilizados	surveyDataControl: DataControl questionOptions: QuestionOptions, answersValidation: AnswersValidationService

Nombre	multiple-answers.component.html
Descripción	Documento HTML que muestra una advertencia (abrir un NgbModal) cuando el usuario desea eliminar una fila de respuestas.

Nombre	question.options.ts
Descripción	Como se han visto en la sección de Datos , existe una amplia variedad de tipos de datos. Entre ellos, los datos desplegables requieren un servicio específico encargado de mapear las opciones y respuestas, con el fin de presentar la información a los usuarios de manera clara y estructurada. Esta solución mejora la experiencia del usuario al garantizar que los datos se muestren en un formato intuitivo y fácil de interpretar.
Atributos	- survey: Survey; - questionsOfSection: Question[];
Métodos	+setSpecialDropdownOfQuestion(question: Question): Question +setSpecialDropdown(influencedQuestions: Question[], controlledQuestion: Question, order: number): QuestionAnswerResponse -getAnswerResponseOfDuplicatedDropdown(question: Question): QuestionOption[] -getAnswerResponseOfUnitItemDropdown(questionUnit: Question, questionDropdown: Question, currentQuestion: Question, order: number): QuestionAnswerResponse -getAnswerResponseOfItemDropdown(currentQuestion: Question, answerResponse: QuestionAnswerResponse, order: number): QuestionAnswerResponse +getResponseInDuplicated(question: Question, response: any): string null -setSigpacDropdown(change: boolean, question: Question, order: number, questionsOfSubsection: Question[]): QuestionOption[] -getAnswerResponseOfCity(question: Question, order: number): QuestionAnswerResponse -getDistance(order: number, influencedQuestions: Question[], question: Question): void -getAnswerResponseOfDropdownReference(question: Question, answerResponse: QuestionAnswerResponse): QuestionAnswerResponse -getDynamicDropdown(questionAnswerResponse: QuestionAnswerResponse, dropdown: any[], value: string): QuestionAnswerResponse

Servicios (privados) utilizados	<ul style="list-style-type: none"> - surveyDataControl: DataControl - sigpacService: SigpacService - getCityService: GetCityService - answerValidationComponent: AnswersValidationService - autoCompleteOperation: AutocompleteOperation - answerService: AnswerService - questionService: QuestionService
---------------------------------	---

Nombre	initialize.currentSection.service.ts
Descripción	<p>Este componente se encarga de inicializar los atributos de subsectionControl y, si corresponde, nonDisplayedSubsection.</p> <ul style="list-style-type: none"> ● subsectionControl: es una lista del tamaño del número de subsecciones con: <ul style="list-style-type: none"> ○ wrongSubsection: controla si existe errores en la subsección. Se determina por erróneo si alguna pregunta de la subsección tiene answerResponse.errorNumber. ○ unsavedSubsection: controla si la subsección tiene alguna pregunta sin guardar. Además, se inicializa con verdadero si la subsección no tiene ninguna respuesta creada (primera vez en entrar a la subsección). ● nonDisplayedSubsection: si la sección es de tipo subsection_election, este atributo se encargará de guardar las subsecciones que no se muestren. <ul style="list-style-type: none"> ○ choosen: decide si la subsección es elegible. ○ subsection: lista de subsecciones del mismo orden para elegir completar.
Atributos	<i>No aplica</i>
Métodos	+ setSubsectionAttributes(section: Section, questionsOfSection: Question[]): Section
Servicios (privados) utilizados	<i>No aplica</i>

Nombre	changeSection.service.ts
Descripción	Este componente se encarga de controlar el cambio de sección/página. Tiene que asegurar que no hay respuestas sin guardar, y en caso contrario avisar al usuario (Swal). Con la alerta, el usuario puede decidir si cambiar de sección/página, no cambiarse o no guardar los datos.
Atributos	<i>No aplica</i>
Métodos	+changeLinkAfterEditAnswer(toSection: boolean, prevSection: Section, destiny: any): void -changeSectionOrRouterLink(toSection: boolean, destiny: any):void
Servicios (privados) utilizados	<ul style="list-style-type: none"> - translate: TranslateService, - router: Router, -dataControl: DataControl -answerSavingService: AnswersSaveDataService

El diagrama de clases (Figura 13) es del componente survey. Sus funciones principales son:

1. Mostrar la lista de usuarios dependiendo del rol del usuario:
 - 1.1. admin: muestra todas las encuestas independientemente del tipo. Sin embargo, solo permite editar las que este haya creado. Además, permite filtrado por las encuestas de tipo “infraestructura” o por las de tipo “cálculo”.
 - 1.2. user: solo se muestran las encuestas de tipo cálculo de las que es autor.
2. Crear nueva encuesta de tipo cálculo.
3. Redirigir a la página de secciones de dicha encuesta (depende del componente section).
4. Progreso del número de secciones/subsecciones completadas.
5. Ordenación por los distintos atributos de la encuesta (id, fecha de creación, nombre...)
6. Tutorial guiado para ayudar a los usuarios a entender el funcionamiento de las encuestas.

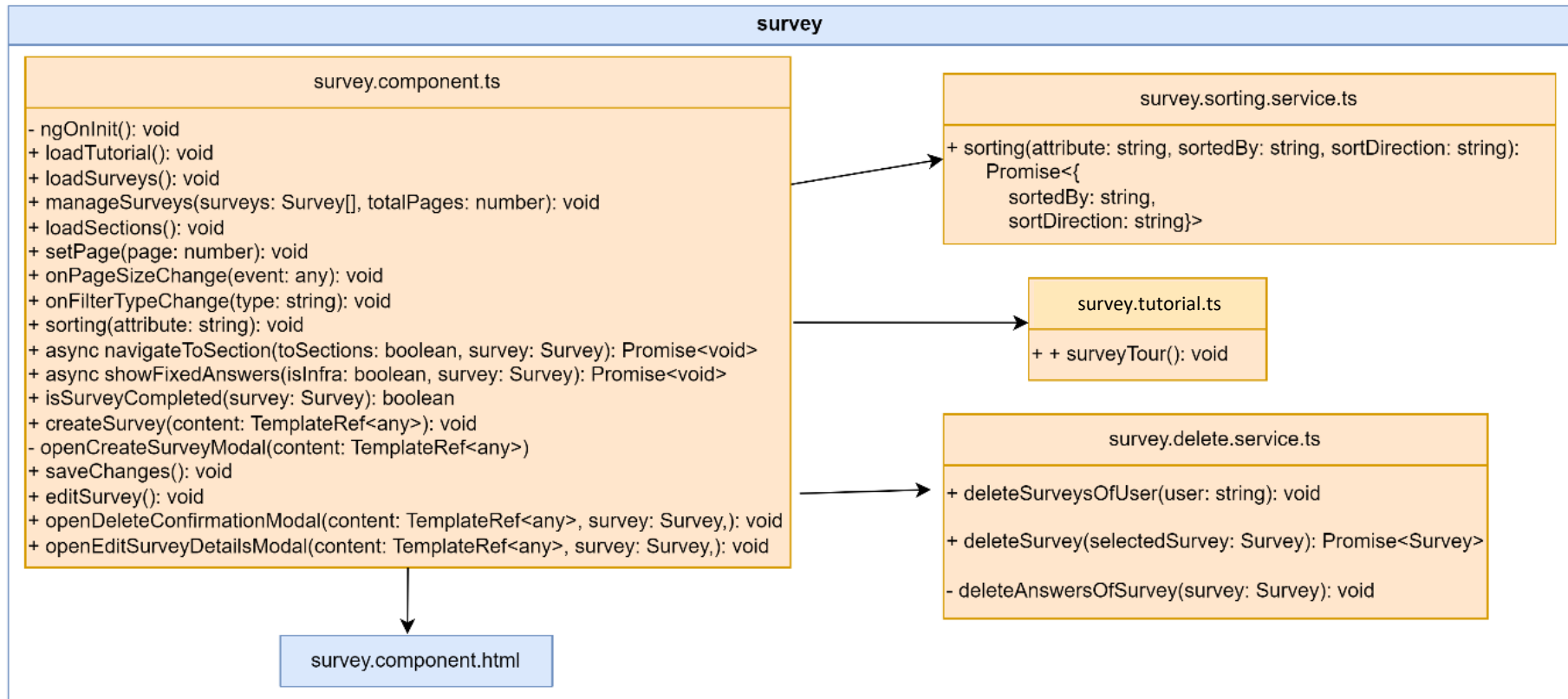


Figura 13: Diagrama de clases (survey)

Nombre	survey.component.ts
Descripción	Este componente se encarga de crear, editar y mostrar las encuestas de los usuarios. Los usuarios normales solo pueden ver sus propias encuestas; mientras que los administradores pueden ver el de todas las personas. Sin embargo, las encuestas solo pueden ser editadas por el usuario que las creó.
Atributos	<ul style="list-style-type: none"> + selectedSurvey: Survey + adminUser: boolean + sections: Section[] + displayedSurveys: Survey[] + years: number[] + surveyYear: number + pageSizes: number[] + pageSize: number + currentPage: number + survPages: number[] + sortDirection: string + sortedBy: string + totalSubsections: number + totalSections: number + totalInfraSubsections: number + totalInfraSections: number + user: User + filterAllList: string[] + filterAll: string
Métodos	<ul style="list-style-type: none"> - ngOnInit(): void + loadTutorial(): void + loadSurveys(): void + manageSurveys(surveys: Survey[], totalPages: number): void + loadSections(): void + setPage(page: number): void + onPageSizeChange(event: any): void + onFilterTypeChange(type: string): void + sorting(attribute: string): void + async navigateToSection(toSections: boolean, survey: Survey): Promise<void> + async showFixedAnswers(isInfra: boolean, survey: Survey): Promise<void> + isSurveyCompleted(survey: Survey): boolean + createSurvey(content: TemplateRef<any>): void - openCreateSurveyModal(content: TemplateRef<any>) + saveChanges(): void + editSurvey(): void + openDeleteConfirmationModal(content: TemplateRef<any>, survey: Survey,): void + openEditSurveyDetailsModal(content: TemplateRef<any>, survey: Survey,): void
Servicios (privados) utilizados	<ul style="list-style-type: none"> modalService: NgbModal, router: Router, userDataService: UserDataService, surveyService: SurveyService, sectionService: SectionService, surveyDelete: DeleteSurvey, surveyFunctionService: SurveyTableFunctionsService, userService: UserService, surveyTutorial: SurveyTutorial, translate: TranslateService

Nombre	survey.table.functions.service.ts
Descripción	Este componente permite ordenar la lista de encuestas ascendente o descendente por los distintos atributos de la encuesta.
Atributos	
Métodos	+ sorting(attribute: string, sortedBy: string, sortDirection: string): Promise<{ sortedBy: string, sortDirection: string}>
Servicios (privados) utilizados	<i>No aplica</i>

Nombre	survey.tutorial.ts
Descripción	Este componente muestra los pasos del tutorial para los usuarios.
Atributos	- surveyDriver: Driver - surveySteps: DriveStep[]
Métodos	+ surveyTour(): void
Servicios (privados) utilizados	- translate: TranslateService

Nombre	survey.delete.ts
Descripción	Este componente sirve para eliminar las encuestas y respuestas del tutorial. Al eliminar un usuario también se eliminarán todas las encuestas y respuestas relacionadas con dicho usuario.
Atributos	+ infraSections: Section[] + sections: Section[]
Métodos	+ deleteSurveysOfUser(user: string): void + deleteSurvey(selectedSurvey: Survey): Promise<Survey> - deleteAnswersOfSurvey(survey: Survey): void
Servicios (privados) utilizados	- surveyService: SurveyService - answerService: AnswerService - sectionService: SectionService

10.4. Diagrama de componentes de sistema

Una vez analizadas las clases y su funcionalidad, es fundamental comprender su interacción dentro del sistema mediante el diagrama de componentes. Este diagrama proporciona una visión clara de la estructura modular del software, las dependencias entre componentes y su integración con elementos externos. A continuación, se detallan los componentes principales que intervienen en el módulo de recolección de datos.

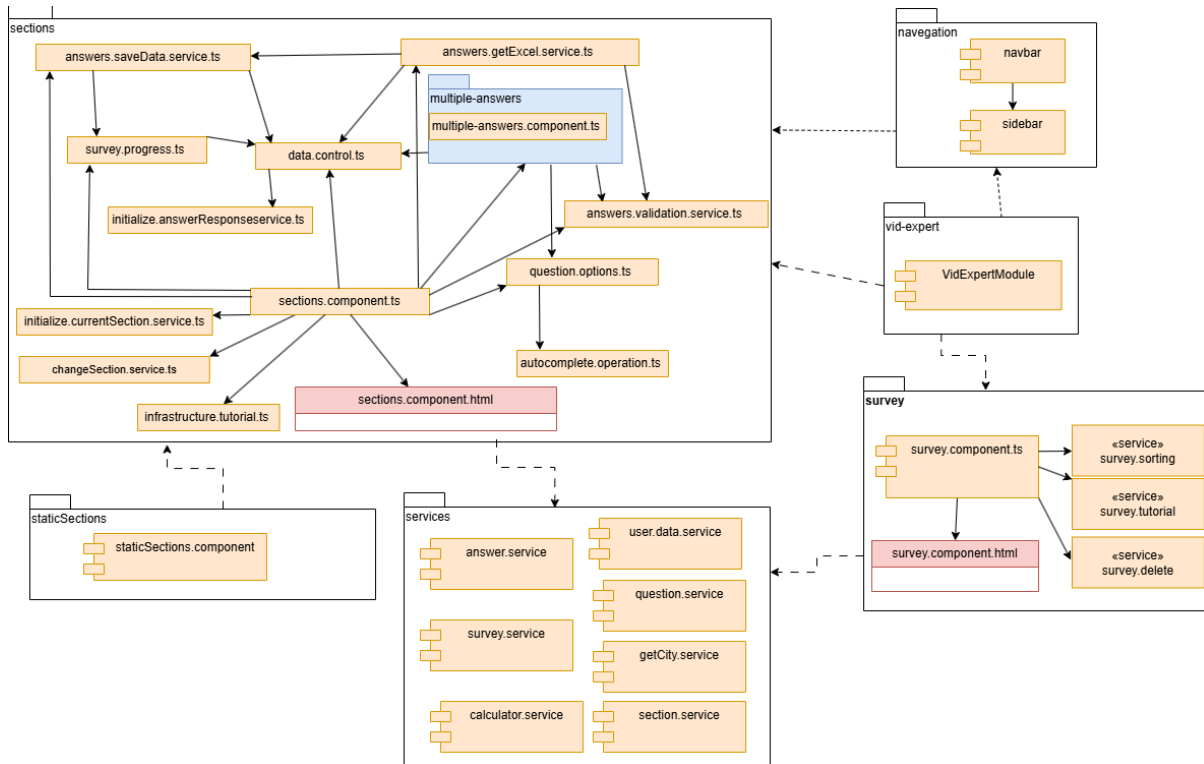


Figura 14: Diagrama de componentes

- sections: componente encargado de mostrar las secciones y preguntas de las encuestas. Implementa la lógica para que los usuarios puedan completar y editar respuestas, así como validar los datos ingresados. Además, gestiona la redirección automática de usuarios según su rol o permisos.
- staticSections: componente que presenta las secciones, preguntas y respuestas en modo de solo lectura, sin permitir modificaciones.
- services: conjunto de servicios responsables de la comunicación entre el backend y el frontend, facilitando el intercambio de datos.
- survey: componente principal que administra la visualización de la lista de encuestas disponibles.
- vid-expert: componente dedicado al manejo de **URLs** y al control de la navegación entre páginas.
- layout: componente encargado de navbar, sidebar, footer. El contenido de la barra de navegación lateral se adapta dinámicamente al ingresar al módulo de recolección de datos. Permite al usuario cambiar de sección al seleccionar una de ellas y muestra en tiempo real el estado de completitud de cada sección. En la [Figura 15](#), se contrasta su apariencia estándar (derecha) con la versión adaptada al módulo (izquierda).

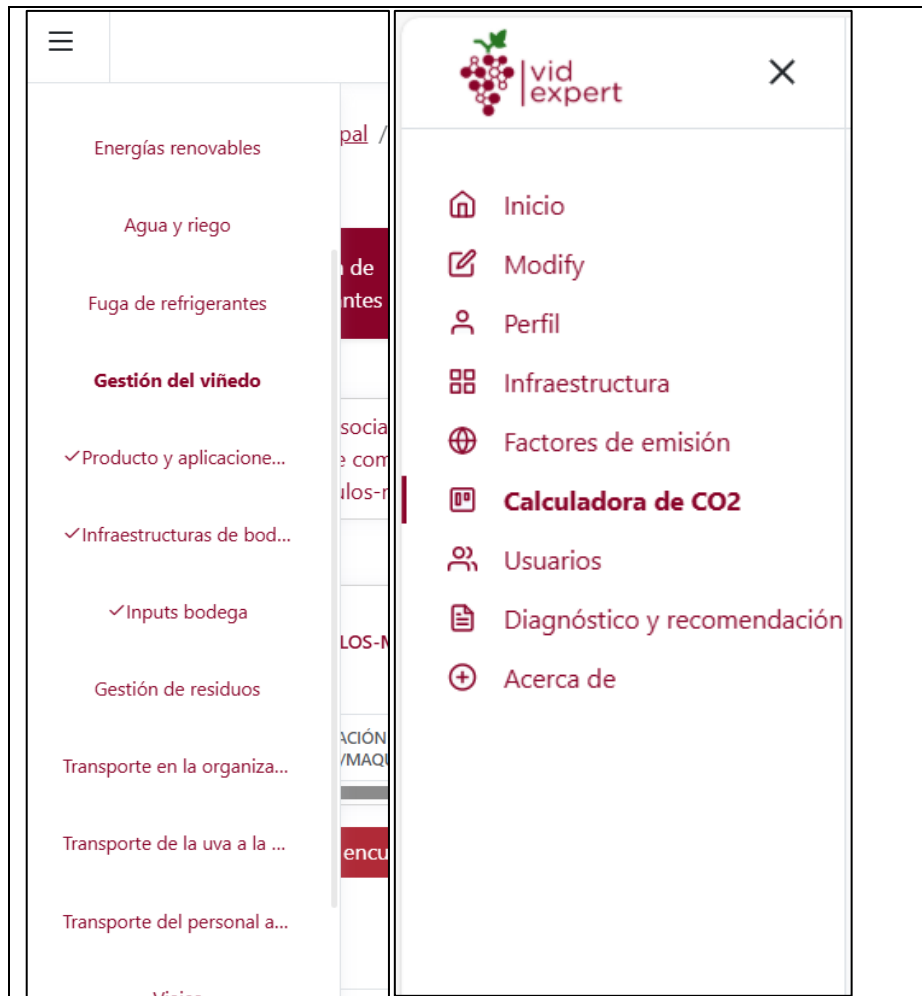


Figura 15: Diferencia entre “section sidebar” y “normal sidebar”

10.5. Diagramas de actividad

En este apartado se analiza el comportamiento del sistema frente a interacciones del usuario, utilizando diagramas de actividad **UML** para modelar de manera clara y estructurada el flujo de procesos. Estos diagramas permiten visualizar las secuencias de acciones, las decisiones tomadas por el sistema y los posibles caminos alternativos, lo que resulta fundamental para comprender la lógica subyacente en la ejecución de funcionalidades clave.

Para facilitar el entendimiento del diagrama de actividad, se va a seguir la siguiente leyenda:

- Inicio de una actividad: indica el punto de partida del flujo.
- Final de una actividad: finalización del flujo
- ▭ Conjunto de sub-actividades: conjunto de tareas subyacentes dentro del flujo
- ➔ Flecha: indica la siguiente acción/decisión
- ◆ Nodo de decisión: condiciones o bifurcaciones del flujo.
- Actividad: acción específica realizada por el sistema

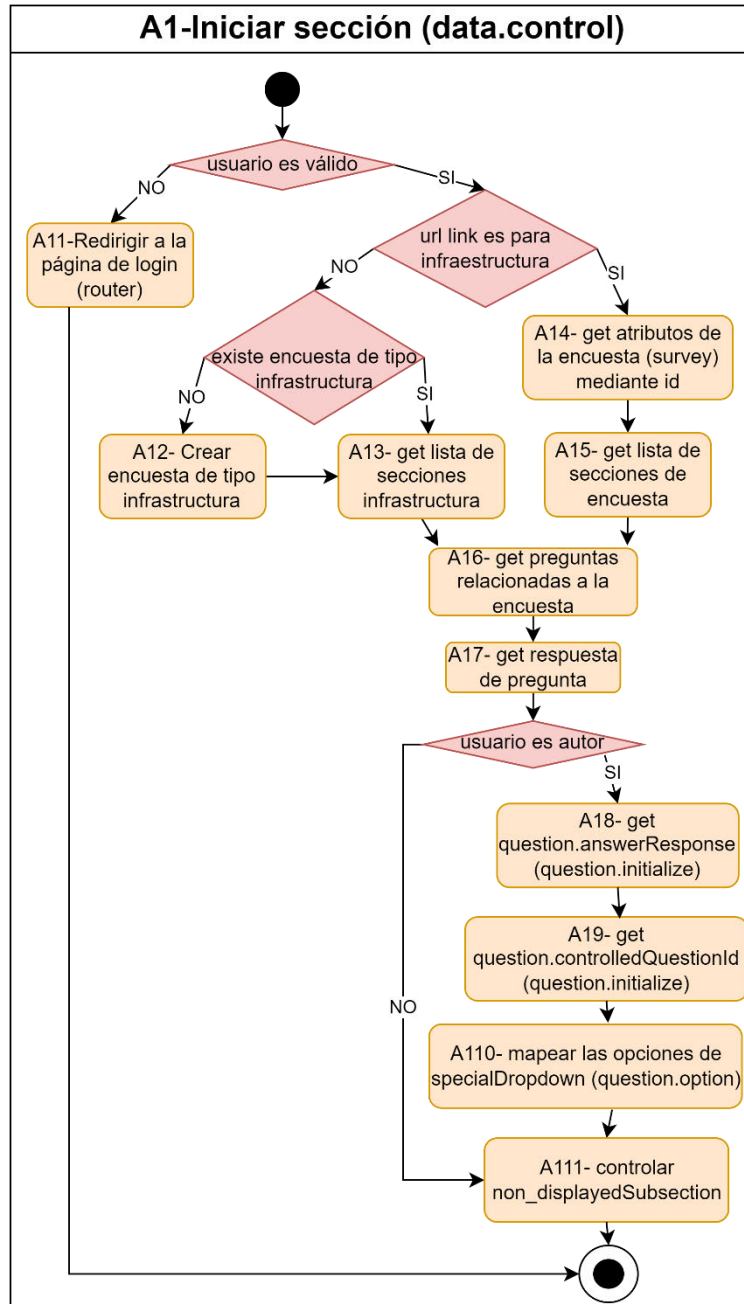
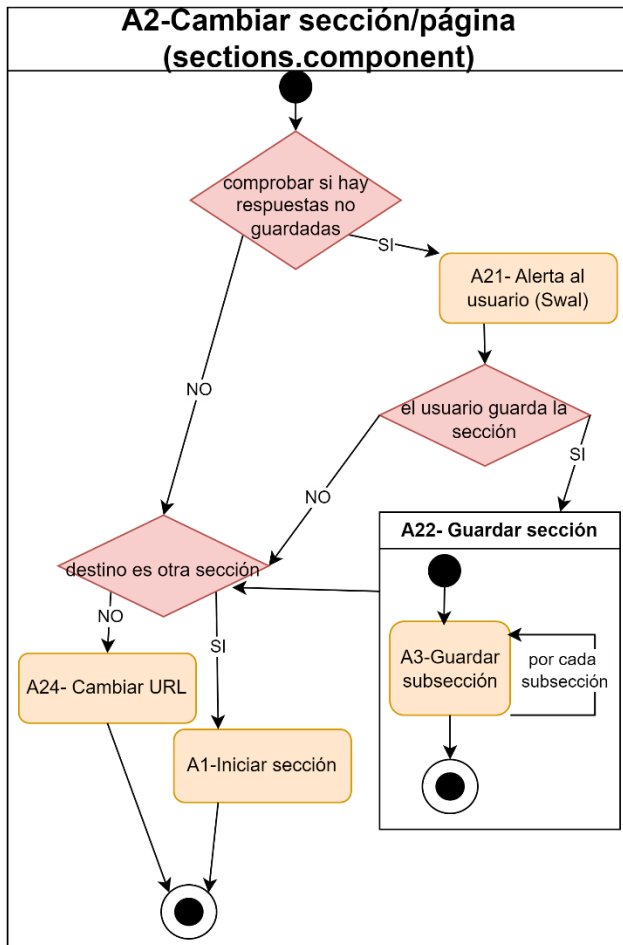


Figura 16: Actividad iniciar sección

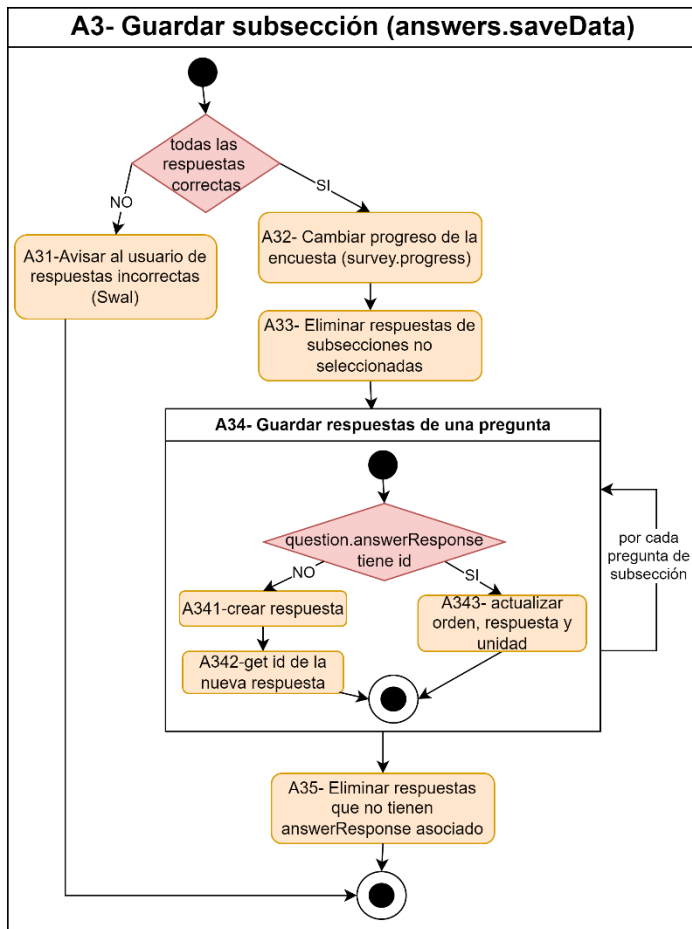
Identificador	A1- Iniciar sección
Requisito de inicio	La actividad se inicializa cuando el usuario entra en una sección.
Descripción	Una sección se refiere a las secciones de una encuesta. La encuesta tiene que existir y ser válida.
Clases involucradas	<ul style="list-style-type: none"> • A11 - A17 data.control • A18, A19 question.initialize • A110 question.option • A111- multiple.answers.component



A21- Alerta al usuario

Figura 17: Actividad cambiar sección/página

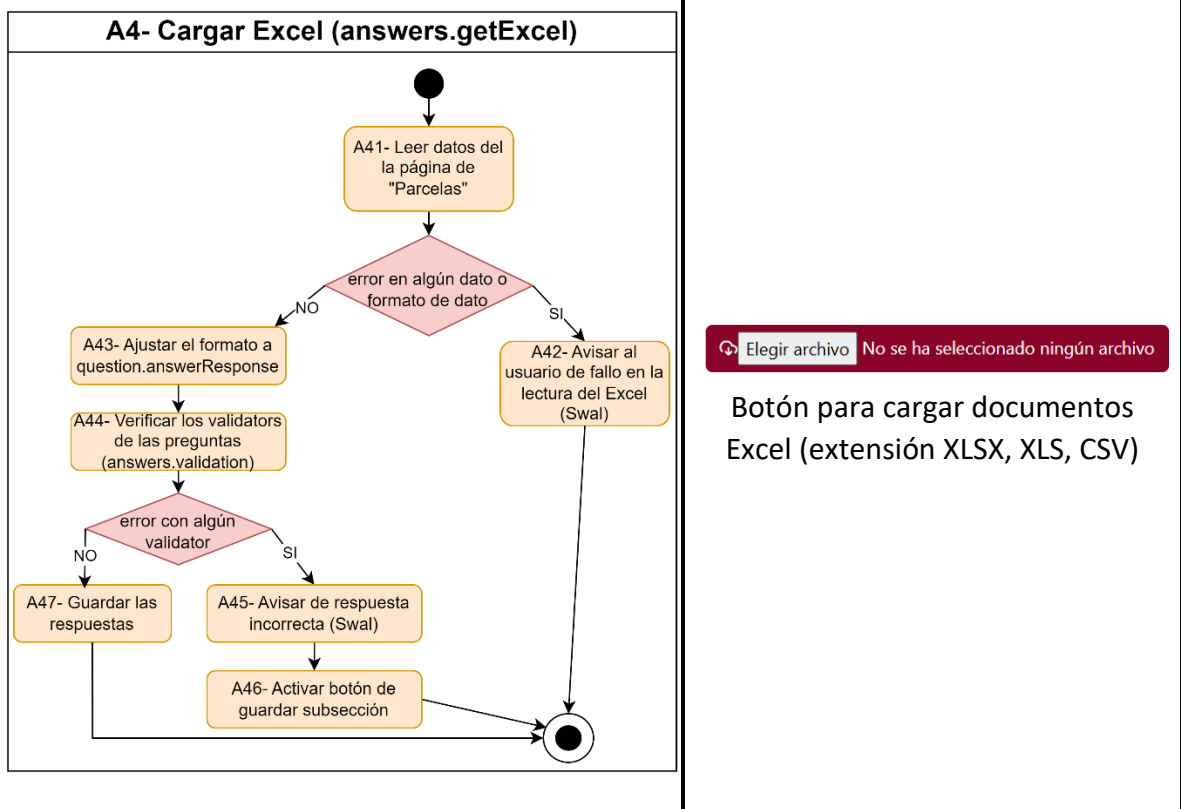
Identificador	A2- Cambiar sección
Requisito de inicio	El usuario está en una sección y desea cambiar a otra
Descripción	Esta actividad utiliza A1-Iniciar sección después de validar los datos y en caso necesario avisar al usuario para decidir el siguiente curso de acción.
Clases involucradas	A21, A24 sections.component A22 answers.saveData.service.ts A1 todas las clases involucradas en A1 A3 todas las clases utilizadas en A3



A31- Avisar al usuario de respuestas incorrectas

Figura 18: Actividad guardar respuestas

Identificador	A3- Guardar subsección
Requisito de inicio	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el botón de guardar subsección. 2. El usuario quiere cambiar y guardar sección cuando hay respuestas sin guardar.
Descripción	Esta actividad describe todos los pasos relacionados con el guardado de las respuestas, con previa validación de todos los question.answerResponse. Se tiene que mapear las respuestas para tener un formato de almacenamiento común y actualizar el progreso de la encuesta.
Clases involucradas	A31, A33, A34 (incluido sub-actividades), A35 answers.saveData.service.ts A32 survey.progres.ts

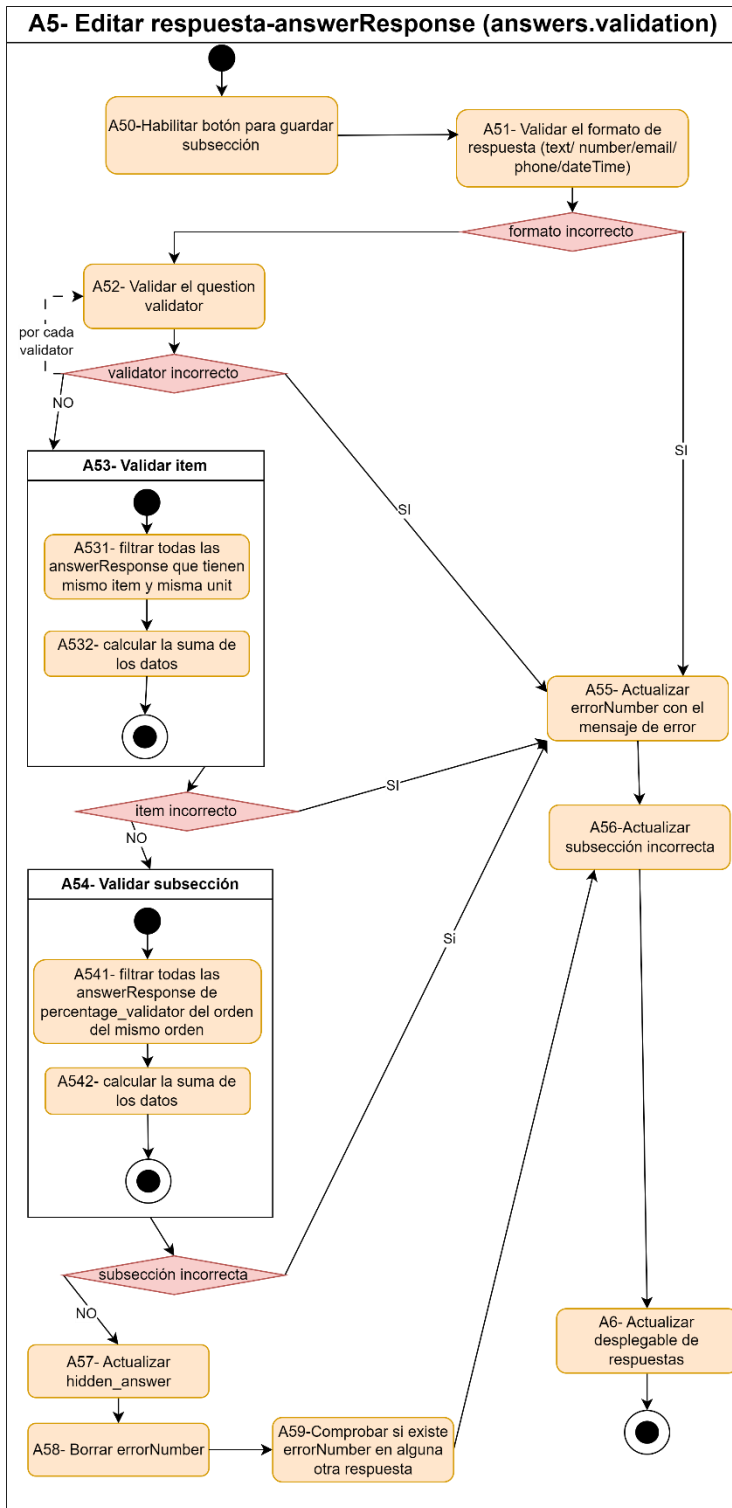


Elegir archivo No se ha seleccionado ningún archivo

Botón para cargar documentos Excel (extensión XLSX, XLS, CSV)

Figura 19: Actividad cargar Excel

Identificador	A4- Cargar Excel
Requisito de inicio	El usuario pulsa sobre el botón para cargar el Excel.
Descripción	El sistema valida el formato del Excel y de los datos para ajustarse a nuestro sistema, carga y valida los datos y los guarda directamente.
Clases involucradas	A41, A42, A45, A46 answers.getExcel.service.ts A43 data.control.ts A44 answers.validation.service.ts A47 answers.saveData.service.ts



#	NOMBRE IDENTIFICATIVO DE LA MAQUINARIA
1	Maquinaria 1
2	text Indique el nombre identificativo de la parcela

Respuesta sin errorNumber
y respuesta con
errorNumber

Figura 20: Actividad de editar respuesta

Identificador	A5- Editar respuesta-answerResponse
Requisito de inicio	El usuario hace una modificación a una de las respuestas.
Descripción	El sistema tiene que verificar los datos en tiempo real por cada cambio introducido por los usuarios.
Clases involucradas	answers.validation.service.ts

A6-Actualizar desplegables de respuestas (question.option)

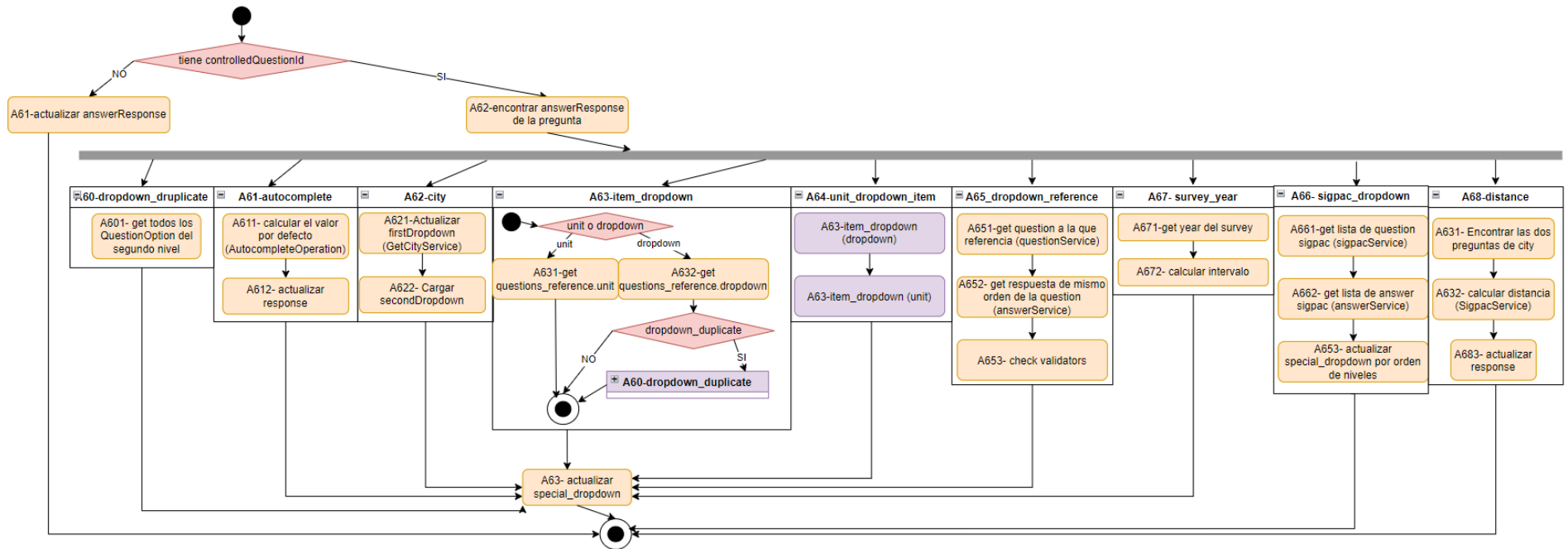


Figura 21: Actividad de actualizar question.option

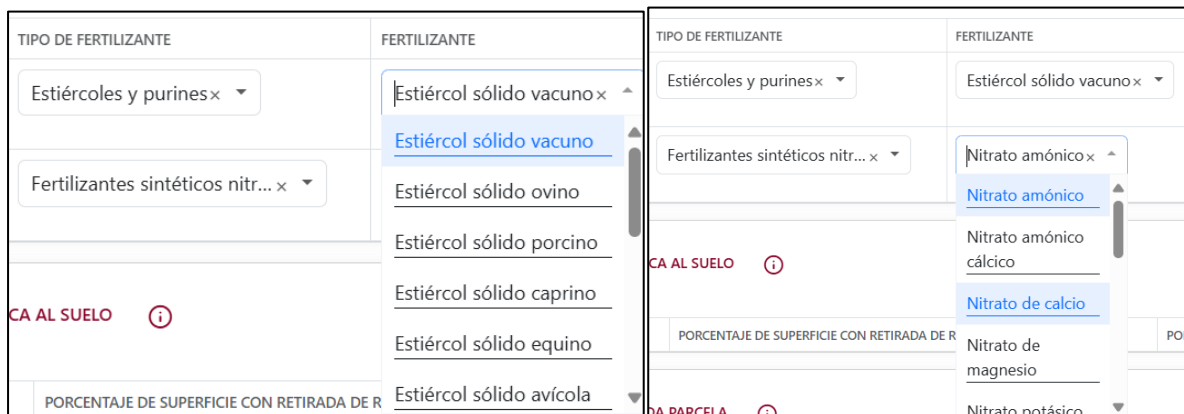


Figura 22: Cambiar el dropdown según el valor de otra pregunta

La Figura 22 muestra que la segunda pregunta (*Fertilizante*) cambia los valores de desplegables dependiendo de la primera pregunta (*Tipo de fertilizante*).

Identificador	A6- Actualizar desplegables de respuestas
Requisito de inicio	El usuario ha modificado una respuesta (answerResponse).
Descripción	Algunas preguntas influyen en la lista de desplegables de otra pregunta. Para ello sirve el atributo de controlledQuestionId, que es el id de la pregunta a la que influye. Si se detecta que la pregunta tiene un controlledQuestionId, se tiene que filtrar el tipo de la pregunta influenciada para tomar acciones en base al formato.
Clases involucradas	question.option

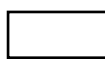
10.6. Comportamiento visual

En este apartado se analizará el comportamiento de la interfaz de usuario (clase html) frente a las acciones del usuario. Se utilizarán actividades del Diagrama de actividad para simplificar el diseño. La notación que se utiliza es la siguiente:

- Inicio de una actividad: indica el punto de partida del flujo.



Final de una actividad: finalización del flujo.



Conjunto de sub-actividades: conjunto de tareas subyacentes dentro del flujo



Flecha: indica la siguiente acción/decisión



Nodo de decisión: condiciones o bifurcaciones del flujo.



Actividad: acción específica realizada por el sistema



Acción: acción realizada por el usuario.

Como se puede apreciar en la [Figura 23](#), una vez que se ha entrado en la página de lista de encuestas, el sistema verifica el rol del usuario para mostrarle una de las dos ventanas principales de la página. En la página del usuario “admin” se muestran las columnas de usuario (email del usuario que ha creado la encuesta) y tipo (si la encuesta es de tipo Infraestructura o Cálculo) que no tiene el usuario normal. Además, el “admin” puede filtrar la lista de encuestas por tipo. A partir de esta primera página, el usuario, independientemente del rol, puede:

- Crear nueva encuesta (Acción 1): [Diagrama_de_casos_de_uso](#) CU3-Crear encuestas. La ventana emergente de “Crear nueva encuesta” se ha implementado con la librería de NgbModal, mientras que “No ha completado la infraestructura” se ha hecho con sweetAlert2.
- Eliminar encuesta existente (Acción 10): [Diagrama_de_casos_de_uso](#) CU4-Eliminar encuestas. La ventana emergente se ha hecho con NgbModal.
- Hacer tutorial (Acción 4): [Diagrama_de_casos_de_uso](#) CU14-Hacer tutorial. Se ha usado driver.js.
- Ver encuesta, solo lectura de respuestas (Acción 7): [Diagrama_de_casos_de_uso](#) CU1-Visualizar encuestas. Se utiliza RouterModule de videxpertModule para cambiar la URL.
- Editar encuesta (Acción 8): [Diagrama_de_casos_de_uso](#) CU5-Editar encuesta. Se utiliza RouterModule de videxpertModule para cambiar la URL.
- Ver resultados del cálculo de Co2 (Acción 9). Se utiliza RouterModule de videxpertModule para cambiar la URL.

10.6.2. Componente section

El diagrama de flujo del componente section es demasiado grande, lo que dificulta la lectura y explicación. Por ello, se explicarán las páginas y ventanas emergentes en distintas figuras.

En la [Figura 24](#) se modela la estructura de una página section o static. Ambas páginas siguen la misma estructura, pero se diferencian en los permisos de edición.

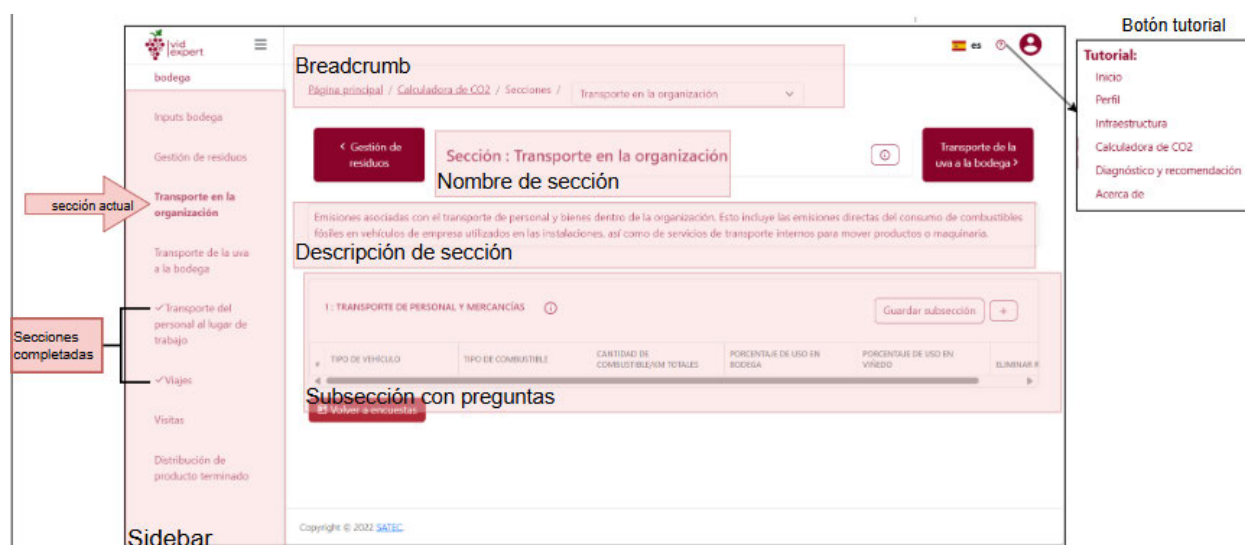


Figura 24: Partes de section o static

- sidebar: barra de navegación lateral. Este apartado pertenece al componente layout.component.ts. A diferencia de lo que ocurre en otras páginas, el sidebar tiene que actualizarse de forma automática en las páginas de tipo sección, dependiendo de las secciones y el progreso de estas, puesto que la lista de secciones de encuestas tipo infraestructura es distinta a las de tipo cálculo.
- breadcrumb: este apartado muestra el camino desde la página inicial hasta la actual. Además, tiene un desplegable de las secciones, haciendo que sea posible navegar a otra.
- Botón tutorial: botón en el navbar para mostrar tutoriales al usuario.

En la [Figura 26](#) se puede ver una página de sección de solo lectura. Las respuestas y unidades se representan directamente como texto en plano y no se puede saber si la sección es subsection_election.

vid expert

es

Página principal / Calculadora.Co2 / Actividades

Sección : Actividades

Volver a editar Ir a cálculos Combustión fija >

Información sobre las actividades llevadas a cabo por la organización, bien sea para la producción de productos enológicos u otras actividades adicionales, como enoturismo.

1: CANTIDAD DE UVA PRODUCIDA

PREGUNTA:	CANTIDAD EMPLEADA EN LA BODEGA DE LA PROPIA EMPRESA PARA ELABORAR VINO PROPIO	CANTIDAD EMPLEADA EN LA BODEGA DE LA PROPIA EMPRESA PARA ELABORAR VINO DE TERCEROS	CANTIDAD DESTINADA A COOPERATIVA/COMPRADORES HABITUALES	CANTIDAD DESTINADA A COMPRADORES AJENOS A LA EMPRESA
1. Uva tinta	0 kg	0 kg	0 kg	0 kg
2. Uva blanca	0 kg	0 kg	0 kg	0 kg

Figura 25: Página de solo lectura

Esta página tiene dos botones que no se han explicado antes:

- Volver a editar: todas las páginas de sección de solo lectura tienen este botón y redirigen a la página para su edición. Sin embargo, solo se redirige con éxito si el usuario es el autor de la encuesta y tiene permiso.
- Ir a cálculo: una vez que la encuesta está completa (tiene todas las secciones completas), este botón redirige a la página donde se muestran los resultados del cálculo de Co2.

Una vez conocidas las partes, vamos a describir las acciones que puede hacer el usuario, el caso de uso relacionado y la respuesta del sistema.

- Acción 1: Cambiar de sección. [Diagrama_de_casos_de_uso](#) CU12-navegar entre secciones. El usuario puede cambiar de sección o página pulsando sobre cualquiera de los botones que señalan las flechas.



Figura 26: Formas de cambiar de sección

- Durante la actividad 2 (**A2- Cambiar sección**), se comprueba si existen respuestas sin guardar. En caso afirmativo, el usuario es alertado según la [Figura 27](#), usando Swal.fire de sweetAlert2.



Figura 27: Alerta de respuestas sin guardar

- Si en la ventana se responde que no se guardan las respuestas, también se notificará al usuario para que éste tenga claro los datos que se han guardado.

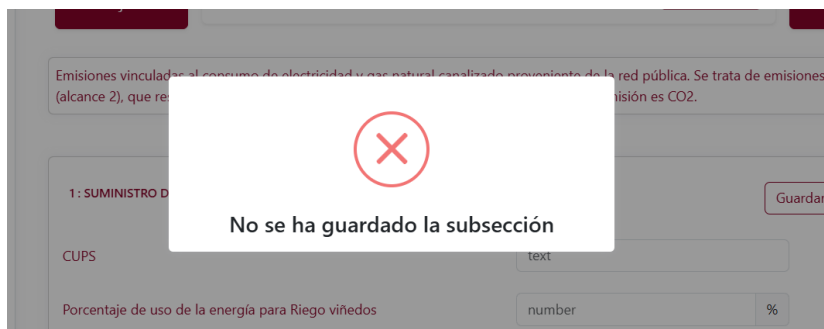


Figura 28: Alerta de subsección no guardado

- Durante la actividad 3 (Emisiones de CO₂
-), se comprueba si existen respuestas erróneas. En caso afirmativo, el usuario es alertado según la Figura 28Figura 29, usando Swal.fire de sweetAlert2

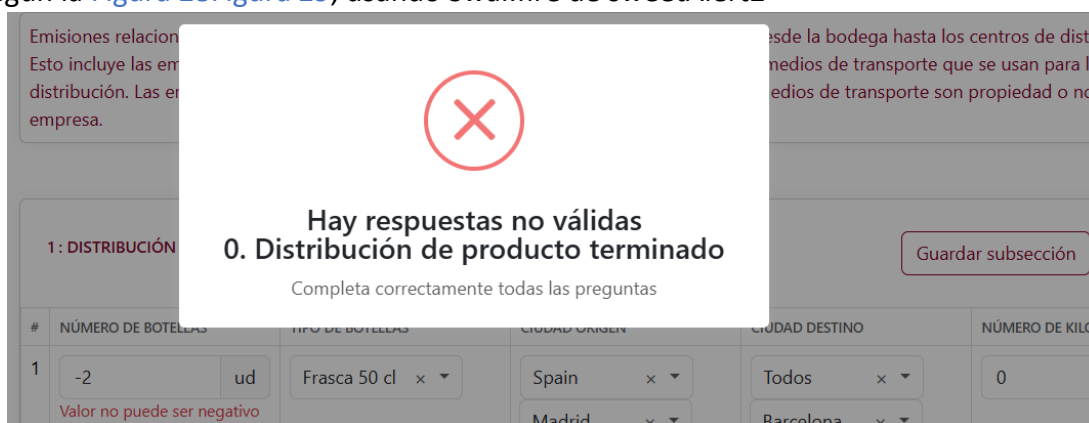


Figura 29: Alerta de respuestas erróneas

- Visualizar detalles de encuesta. Este botón permite al usuario ver los detalles de la encuesta que está completando, así como la sección actual y sus subsecciones, y si ésta está completa. También permite moverte a la sección que pulsas.



Figura 30: Detalles de encuesta

7. Tutorial: [Diagrama_de_casos_de_uso](#) CU14-Hacer tutorial. Se ha implementado con driver.js

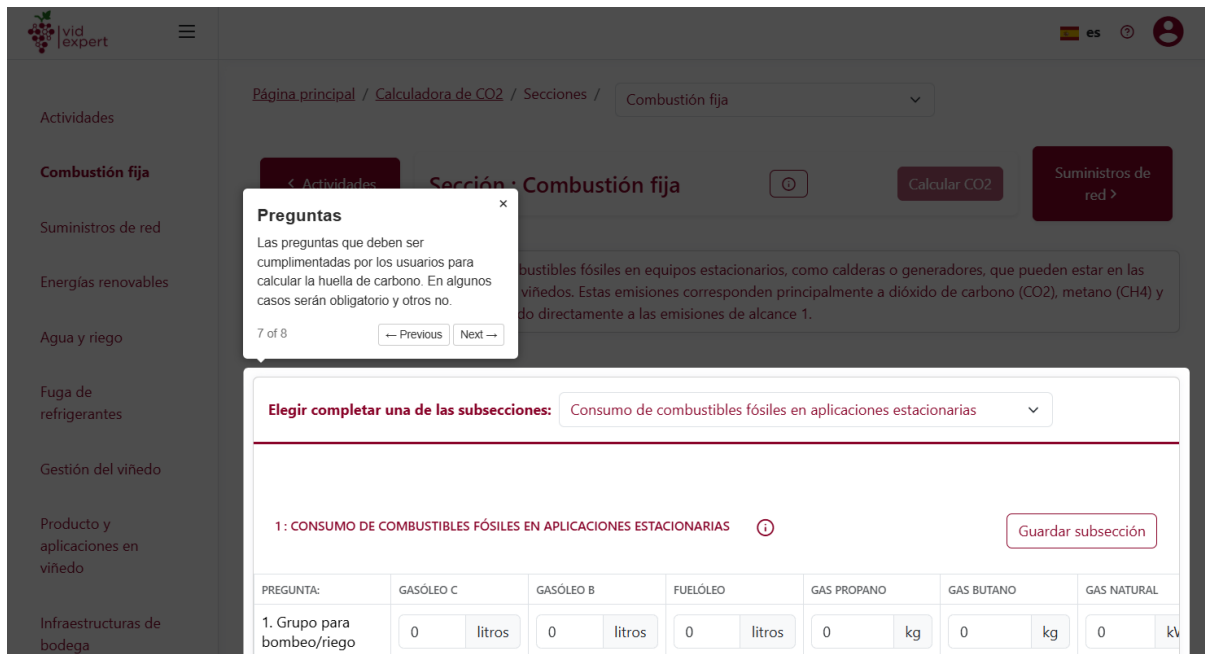


Figura 31: Tutorial section

8. Elegir entre completar secciones: [Diagrama_de_casos_de_uso](#) CU13-elegir entre subsecciones a completar. Las secciones de subsection_election permiten elegir completar entre varias subsecciones. Por ello, las respuestas de la subsección que no ha sido seleccionada tienen que ser eliminadas para mantener la consistencia de los datos.



Figura 32: Atributo "subsection_election"

- Una vez completada la encuesta, aparece el botón de calcular CO2 para calcular las **Emisiones** de gases de efecto invernadero producidos durante la actividad vitivinícola ([Diagrama_de_casos_de_uso CU11-hacer cálculo de Co2](#)) Esta acción puede durar varios segundos (máximo 30s), por lo que para evita incongruencias, se deshabilita todas las acciones del usuario y aparece un mensaje de carga.

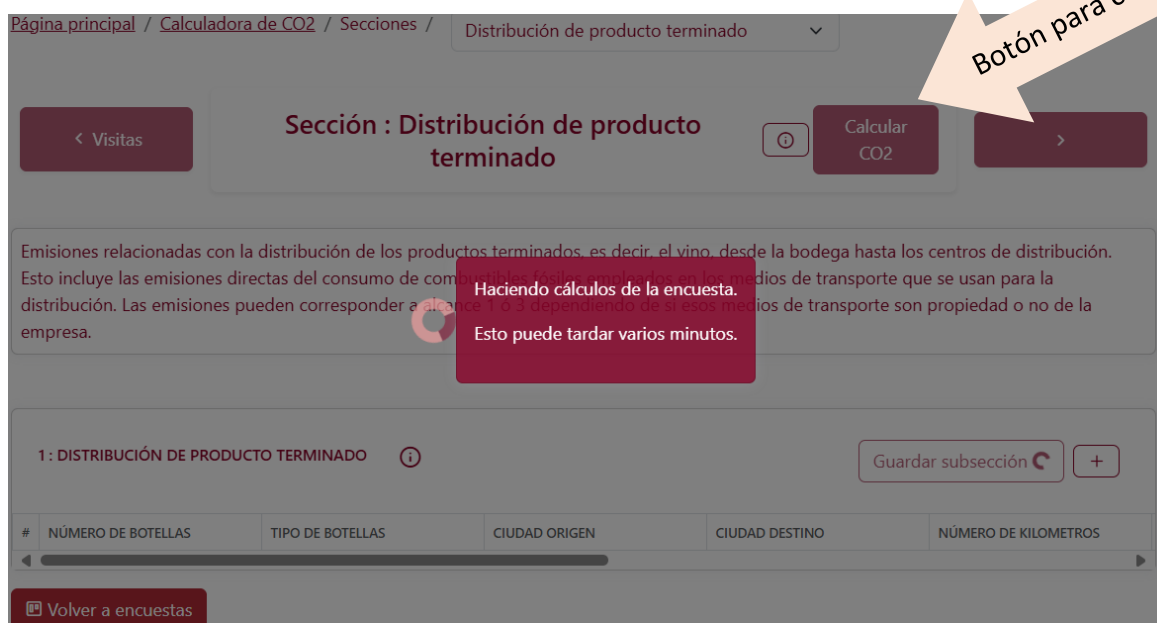


Figura 33: Carga para hacer cálculos

- Un comportamiento interesante es el de hidden. Este atributo aparece dentro de un validator. Si se cumple la condición, la pregunta ocultará y borrará la respuesta. En la [Figura 34](#) se puede ver que las respuestas “No” en la pregunta “SI/NO” hacen que se oculten las respuestas de “Climatizado”.

1: INSTALACIONES DE LA EMPRESA				Guardar subsección	
PREGUNTA:	SI/NO	CLIMATIZADO	ESPACIO CONSTRUIDO		
1. Espacios de elaboración	No x ▾		0		m2
2. Espacios de crianza envejecimiento	Sí x ▾	Sí x ▾	0		m2
3. Botellero (si es diferente de la sala de crianza)	Sí x ▾	Sí x ▾	0		m2
4. Salas de embotellado	No x ▾		0		m2

Figura 34: Comportamiento de hidden

11. Implementación

En la clase `videxpert.module.ts` del componente `videxpert` se utiliza rutas anidadas (`children` en la figura [Figura 35](#)). Esto permite agrupar rutas relacionadas bajo un mismo contexto (el especificado en `path`) y evita que los componentes hijo se salgan del espacio de interfaz gráfica designada al padre.

Los atributos mostrados muestran:

- **path** es la ruta del componente. Se anida el valor a la ruta del padre o en su defecto con la ruta base (`"http://"`).
- **component** es el nombre del componente que tiene que ser inicializado. De ella se obtiene la lógica (archivo Typescript), la interfaz gráfica (archivo HTML) y los estilos (archivo SCSS).
- **canActivate** es un conjunto de componentes que pueden activarse como servicios. El que se usa en la imagen [Figura 35](#) es un componente de autorización de usuarios.
- **children** es una lista de rutas anidadas con sus respectivos atributos. En el primer com

```
{
  path: 'survey/section/:surveyId',
  component: SectionsComponent,
  canActivate: [AuthGuard],
  children: [
    {
      path: ':sectionCode',
    }
  ],
},
{
  path: 'infrastructure',
  component: SectionsComponent,
  canActivate: [AuthGuard],
  children: [
    {
      path: ':sectionCode',
    }
  ],
},
```

Figura 35: Rutas anidadas

El servicio `data.control.service.ts` ([Figura 36](#)) del componente `section` emplea `BehaviorSubject` de `RxJS` para garantizar el control unidireccional del flujo de datos, integridad de la información y reactividad en tiempo real.

Los otros componentes se subscriben al atributo (Observable) que necesitan y cada vez que se cambia el valor, los subscriptores reciben el cambio.

```
private user: User;
private survey: Survey;
private surveySubject : BehaviorSubject<Survey | null> = new BehaviorSubject<Survey | null>(_value: null);
survey$ : Observable<Survey | null> = this.surveySubject.asObservable();
private sectionsSubject : BehaviorSubject<Section[]> = new BehaviorSubject<Section[]>(_value: []);
sections$ : Observable<Section[]> = this.sectionsSubject.asObservable();
private currentSectionSubject : BehaviorSubject<Section | null... = new BehaviorSubject<Section | null>(_value: null);
currentSection$ : Observable<Section | null> = this.currentSectionSubject.asObservable();
private questionsSubject : BehaviorSubject<Question[]> = new BehaviorSubject<Question[]>(_value: []);
questionsOfSection$ : Observable<Question[]> = this.questionsSubject.asObservable();
private initialQuestionsSubject : BehaviorSubject<Question[]> = new BehaviorSubject<Question[]>(_value: []);
initialQuestions$ : Observable<Question[]> = this.initialQuestionsSubject.asObservable();
private answersFromSection: Answer[];
private answersSubject : BehaviorSubject<Answer[]> = new BehaviorSubject<Answer[]>(_value: []);
answersOfSection$ : Observable<Answer[]> = this.answersSubject.asObservable();
```

Figura 36: BehaviorSubject de data.control

En la clase data.control.ts del componente section se hace la redirección del usuario en base al URL (Figura 37).

```
private getUrlLink(router: Router) : void { Show usages  xinyao.zhang
const baseUrl : string = router.url;
if (baseUrl.startsWith('/vid-expert/survey/section')) {
  const segments : string[] = baseUrl.trim().split(separator: '/').filter(segment : string => segment !== '');
  this.urlLink = '/' + segments.slice(0, 4).join('/');
  this.getAllSections(segments[3]);
} else if (this.router.url.startsWith('/vid-expert/viewSections')) {
  const segments : string[] = baseUrl.trim().split(separator: '/').filter(segment : string => segment !== '');
  this.urlLink = '/' + segments.slice(0, 3).join('/');
  this.getAllSections(segments[2]);
} else if (baseUrl.startsWith('/vid-expert/viewInfrastructure')) {
  const segments : string[] = baseUrl.trim().split(separator: '/').filter(segment : string => segment !== '');
  this.urlLink = '/' + segments.slice(0, 3).join('/');
  this.getAllInfra(segments[2]);
} else if (baseUrl.startsWith('/vid-expert/infrastructure')) {
  const segments : string[] = baseUrl.trim().split(separator: '/').filter(segment : string => segment !== '');
  const link : string = segments.slice(0, 2).join('/');
  this.urlLink = '/' + link;
  this.getAllInfra();
}
}
```

Figura 37: Redirección en base al URL

En la clase general de estilos style.scss del proyecto se ha importado los estilos de driver.js para hacer tutoriales (Figura 38). Estos estilos permiten resaltar el componente al que se hace la referencia del tutorial. En la Figura 39 se puede ver un componente resaltado, el paso actual y el total de pasos del tutorial. Además, en la barra de navegación (navbar) hay un botón-tutorial que navega y muestra el tutorial del componente que se requiere (Figura 40).

```

47 @import "@mdi/font/css/materialdesignicons.min.css";
48 @import "perfect-scrollbar/css/perfect-scrollbar.css";
49 @import "highlight.js/styles/googlecode.css";
50 @import "ngx-owl-carousel-o/lib/styles/prebuilt-themes/";
51 @import "ngx-owl-carousel-o/lib/styles/prebuilt-themes/";
52 @import "animate.css/animate.min.css";
53 @import "@ng-select/ng-select/themes/default.theme.css";
54 @import "metismenujs/scss/metismenujs";
55 @import "dropzone/dist/min/dropzone.min.css";
56 @import "quill/dist/quill.snow.css";
57 @import "angular-archwizard/styles/archwizard.css";
58 @import "simple-datatables/dist/style.css";
59 @import "@swimlane/ngx-datatable/index.css";
60 @import "@swimlane/ngx-datatable/themes/bootstrap.css";
61 @import "@swimlane/ngx-datatable/assets/icons.css";
62
63 // VidExpert styles for LTR
64 @import "./assets/scss/style";
65
66
67 // styles for driver.js for making tutorials
68 @import "driver.js/dist/driver.css";
69
70

```

Figura 38: Página de estilos de la aplicación style.scss

Lista de cuestionarios

Aquí se listan todos los cuestionarios que se están en progreso o que se han finalizado. Presenta varias acciones que permite editar o eliminar los formularios, así también, muestra el progreso de estos.

1 of 10 ← Previous Next →

#	DETALLES	%	PERIODO DE CÁLCULO	%	USUARIO
1	a a		2025		admin@adr
2	Infraestructura		2025		admin@adr

Figura 39: Ejemplo de tutorial (paso 1 de 10)

Botón tutorial

Tutorial:

- Inicio
- Perfil
- Infraestructura
- Calculadora de CO2
- Diagnóstico y recomendación
- Acerca de

Figura 40: Botón de tutorial en navbar

12. Pruebas y Validación

En esta sección se detallan las pruebas realizadas para validar el funcionamiento del sistema.

12.1. Pruebas unitarias

Estas pruebas se enfocan en analizar pequeñas unidades de código asegurando el funcionamiento individual de cada parte. Se han hecho pruebas unitarias para introducción de datos, es decir, pruebas de las entradas de respuestas por pregunta tipo:

1. Tipo de pregunta: por cada valor `question.type` se ha hecho una prueba que verifique que la entrada de datos se limita al tipo de entrada.
2. `validators` por `question`: se ha validado que las respuestas siguen las restricciones que se han puesto según `validators.name`.
3. `validators` por `item`: se ha validado que las respuestas de preguntas con `item` y misma `unit` que `item.validators.unit` siguen las restricciones que se han puesto según `item.validators.validator`.
4. `validators` por `subsection`: se ha validado que las respuestas de las preguntas de `percentage_validator` siguen las restricciones que se han puesto (suma de porcentajes no mayor a 100).

Estas pruebas se han ejecutado directamente sobre el sistema de prueba (`localhost`), puesto que la existencia de un error de entrada de datos solo se muestra al usuario, impidiendo a éste el guardado de las respuestas.

12.2. Pruebas de interfaz gráfica

Estas pruebas verifican el comportamiento integral del sistema desde la perspectiva del usuario final, asegurando la correcta interacción con todos los componentes de la interfaz. Esto sirve para garantizar que la interfaz responde según lo esperado en condiciones normales y críticas, mejorando la experiencia de usuario y la fiabilidad del sistema. Se han ejecutado pruebas sobre:

- Navegación entre páginas/section:
 - Validación de rutas y redireccionamientos entre secciones.
 - Carga correcta de componentes al cambiar de vista.
 - Alerta al usuario en caso de respuestas sin guardar.
- Autenticación de usuario: todas las páginas deben verificar la identidad de usuario y restringir el acceso de las rutas.
- Alerta: los mensajes de alerta (error o éxito al guardar subsección) saltan en los momentos indicados.
- Conexión con backend:
 - GET para recuperación de datos (preguntas, respuestas guardadas).
 - POST/PUT de nuevas respuestas.
 - DELETE de respuestas eliminadas o vacías.

- Tutorial: se ha verificado que el tutorial funciona correctamente cuando el usuario lo requiere.

Para ejecutar las pruebas se ha aplicado en un entorno real sobre el sistema de prueba (en localhost).

12.3. Pruebas de aceptación del cliente

Las pruebas de aceptación son realizadas por el cliente (posible usuario final) para validar que el sistema cumpla con los requisitos funcionales y de negocio establecidos. Se han validado:

- Casos de uso: se ha validado que los casos de uso son los establecidos en el [Diagrama_de_casos_de_uso](#) y el flujo de interacción son los descritos en las tablas de Casos de uso extendido.
- Experiencia de usuario: se comprobó que el sistema fuera fácil de usar, que las interfaces fueran intuitivas y que los tiempos de respuesta fuesen adecuados. También hemos recibido feedback mediante correo de posibles mejoras que se tendrán en cuenta en un futuro.

13. Conclusión

En este apartado se discutirá sobre resultados obtenidos, futuras mejoras, limitaciones e impacto de nuestro módulo de recolección de datos.

13.1. Análisis de resultados

El desarrollo de VIDEXPERT ha demostrado ser una solución viable para el cálculo de la huella de carbono en el sector vitivinícola, logrando:

- Mayor precisión en los cálculos: validación de datos y factores de emisión actualizables.
- Interfaz intuitiva: se resolvió un caso de reclamación, lo que implica pasar de 93% de satisfacción de los stakeholders a un 100% de satisfacción tras el primer uso.
- Integración efectiva: el módulo del frontend y backend comunican datos sin errores en el 99% de los casos evaluados.
- Actualización dinámica de preguntas y factores de emisión sin requerir modificaciones en el código base. La información de las preguntas, factores de emisión y resultados de cálculo se almacenan en la base de datos, lo que permite cambiar las preguntas y ajustarse a los cambios del gobierno y de nuevas leyes sin necesidad de actualizar y desplegar de nuevo la herramienta.

Respecto de los objetivos planteados en 4.1. Objetivos del proyecto, se han completado con éxito el diseño e implementación del módulo de recolección de datos de VIDEXPERT y se han desarrollado los pasos definidos:

1. Desarrollar una interfaz gráfica intuitiva y amigable utilizando **Angular**:
 - a. Tutorial explicativo para todas las páginas de la aplicación.
 - b. Distintas formas de cambiar de sección.
 - c. Reducción de las formas en las que se muestran las preguntas.
2. Diseño de un modelo de datos que sean válidos para mapear distintos formatos de datos:
 - a. Tipos básicos de datos.
 - b. Desplegables de distintos tipos para ajustarse a las necesidades de los stakeholders.
 - c. Relación entre distintas preguntas mediante *questions_reference*.
 - d. Validación a nivel de *question*, *item* y *subsection*.
3. Mapeo de las preguntas proporcionadas por los expertos al formato diseñado:
 - a. Mapear las preguntas proporcionadas por los expertos del sector al formato diseñado.
4. Implementar un sistema de control y validación de los datos para garantizar la calidad y consistencia de la información introducida:
 - a. Distintos tipos y niveles de validación de los datos.
 - b. Mensajes de alerta en caso de error.
5. Creación y almacenamiento de las respuestas recopilados en un formato homogéneo para su posterior procesamiento:
 - a. Clase **Answers** almacena las respuestas en un formato homogéneo.

6. Integrar el módulo de recolección de datos con los demás componentes de VIDEXPERT:
 - a. El módulo de recolección de datos funciona dentro de la herramienta VIDEXPERT (<https://videxpert.benderlab.es/vid-expert>)

Para complementar el análisis de resultados se han analizado algunos indicadores que pueden dar una idea del alcance y complejidad de los trabajos realizados:

- Número de páginas de entrada de datos (secciones):
 - Tipo infraestructura: 4
 - Tipo cálculo: 17
- Número de campos de entrada de datos (question):
 - 57 con item (existe un total de 170 item)
 - 179 sin item
 - Total de campos de entrada de datos: 284 (variación del número de item de las distintas preguntas)
- Campos con validación:
 - question: 178/236 ~ 75%
 - item: 16/170 ~ 9%
 - subsection: 7/48 ~ 14%
 - Total de campos de validación: 44%
- % de páginas de la aplicación con ayuda: 100% tienen tutorial en línea.
- Número de clases:
 - Componente survey: 7
 - Componente section:
 - ◆ Clases para secciones de rellenar respuestas: 6
 - ◆ Clases para secciones de muestra de respuestas estáticas: 4
 - ◆ Clases para secciones compartidas: 13
 - Otros componentes de ayuda:
 - ◆ Modelar datos: 7
 - ◆ Navegación y tutorial: 3
 - ◆ Estilos generales: 1

13.2. Limitaciones y posibles mejoras

Se detallan las principales limitaciones encontradas durante el desarrollo del proyecto:

- Dependencia de datos manuales. La falta de integración con sensores IoT en esta fase requiere la introducción manual de datos, lo que puede introducir variaciones en los cálculos de emisiones. Esto se podría resolver para futuras extensiones.
- Encuestas extensas. El diseño actual incluye cuestionarios con más de 1000 preguntas, lo que puede resultar tedioso para los usuarios que solo deseen calcular emisiones en etapas específicas del proceso. Sin embargo, este inconveniente es un requisito importante para garantizar la precisión en los cálculos.

Una vez descritas las limitaciones, se propondrán mejoras para futuras versiones:

- Automatización de entrada de datos: integración con sistemas ERP y sensores IoT para recopilar datos automáticamente, reduciendo la intervención manual.
- Inteligencia Artificial: implementación de modelos predictivos para optimizar recomendaciones de reducción de emisiones. En esta versión inicial, las recomendaciones son básicas, pero se planea incorporar algoritmos de IA para análisis de coste-beneficio y personalización.
- Certificaciones **MITECO**: vinculación directa con el Registro Nacional de Huella de Carbono para emitir certificaciones desde la plataforma. Gracias a colaboraciones con entidades como la Federación Española del Vino (FEV), se podrá explorar esta posibilidad, agilizando trámites para el sector y las autoridades.

13.3. Impacto

El proyecto VIDEXPERT se alinea estratégicamente con los Objetivos de Desarrollo Sostenible **ODS** de la ONU, generando impactos en cuatro ámbitos clave:

ODS 9 (Industria, innovación e infraestructura).

VIDEXPERT representa una transformación digital al sustituir los métodos tradicionales (documentos en Excel y Word) por un sistema integrado que valida datos de entrada en tiempo real mediante reglas modificables. Además, el sistema está desarrollado mediante una arquitectura de microservicios, lo que permite la fácil integración de futuros sensores IoT y sistemas ERP.

ODS 12 (Producción y consumo responsables).

VIDEXPERT mejora la sostenibilidad mediante las recomendaciones detalladas para la optimización del uso de recursos como agua, energía, productos químicos o combustibles de los vehículos. Además, el análisis detallado de las emisiones de CO₂ durante distintas fases de producción del sector vitivinícola ayuda a los agricultores a conocer los procesos que más influyen en su huella de carbono.

ODS 13 (Acción por el clima)

VIDEXPERT ofrece herramientas avanzadas para calcular la huella de carbono y reducir emisiones de CO₂ mediante algoritmos que calculan tanto emisiones directas (combustibles, fermentación) como indirectas (transporte, energía comprada). Los usuarios, teniendo en cuenta los resultados de las encuestas, podrán evaluar y planificar mejoras propias o utilizar las recomendaciones que el sistema se les ofrece.

ODS 17 (Alianzas para lograr los objetivos)

El desarrollo de VIDEXPERT ha implementado un modelo de trabajo colaborativo e inclusivo, estableciendo un modelo de trabajo de cooperación entre universidades, empresas privadas e instituciones gubernamentales. Este enfoque ha permitido integrar perspectivas

multidisciplinares, desde la investigación académica hasta la aplicación práctica en bodegas, asegurando que la solución tecnológica sea tanto innovadora como adaptada a las necesidades reales del sector vitivinícola. El ODS 17 promueve este tipo de alianzas, que involucren a todos los stakeholders desde las fases iniciales del proyecto y se compartan conocimientos entre instituciones con diferentes enfoques.

14. Bibliografía

- [1] *Introduction. What is Angular?* Accedido: 2025-04-20. URL: <https://angular.dev/overview>
- [2] *Get started with Bootstrap.* Accedido: 2025-05-06. URL: <https://getbootstrap.com/docs>
- [3] *IntelliJ IDEA. The IDE for Professional Development.* Accedido: 2025-05-06. URL: <https://www.jetbrains.com/idea/>
- [4] *MongoDB Atlas: Cloud Document Database.* Accedido 2025-05-05 <https://cloud.mongodb.com/v2/6613cc55157f456e5bc31703#/overview>
- [5] *Metodologías ágiles* <https://www.aden.org/business-magazine/metodologias-agiles/>
- [6] *Qué es scrum y cómo empezar* <https://www.atlassian.com/es/agile/scrum>
- [7] *The SCRUM GUIDE (páginas 273-277)* <https://scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf>
- [8] *Normativa de huella de carbono. Disposición 3379 del BOE núm. 77 de 2014* <https://www.boe.es/boe/dias/2014/03/29/pdfs/BOE-A-2014-3379.pdf>
- [9] *Qué son los certificados MITECO y cómo obtenerlos* <https://marketing4ecommerce.net/certificados-miteco/>
- [10] *Registro de huella de carbono, compensación y proyectos de absorción de dióxido de carbono* <https://www.miteco.gob.es/es/cambio-climatico/temas/registro-huella.html>
- [11] *Federación Española de Vino* <https://www.fev.es/es/>
- [12] *Sustainable Wineries for Climate Protection* <https://www.nqa.com/es-es/certification/standards/Sustainable-Wineries-for-Climate-Protection>
- [13] *Proyecto VIDEXPERT* https://www.fev.es/fev/sostenibilidad-y-responsabilidad/proyecto-vid-expert_995_1_ap.html
- [14] *El Grupo SATEC finaliza el desarrollo de su proyecto Videxpert para reducir la huella de carbono en la viticultura* https://agroinformacion.com/el-grupo-satec-finaliza-el-desarrollo-de-su-proyecto-videxpert-para-reducir-la-huella-de-carbono-en-la-viticultura/#goog_rewarded
- [15] *WINEKIA. Herramienta online para calcular la Huella de Carbono en el sector vitivinícola* <https://www.vinetur.com/2015071420236/herramienta-online-para-calculer-la-huella-de-carbono-en-el-sector-vitivinicola.html>
- [16] *Web winekia* <https://winekia.com/>
- [17] *What is Java Spring Boot.* <https://www.ibm.com/topics/java-spring-boot>
- [18] *Requerimientos Funcionales y No Funcionales: La Guía Completa* <https://byspel.com/requerimientos-funcionales-y-no-funcionales-la-guia-completa/>
- [19] *Todos los diagramas UML. Teoría y ejemplos* <https://diagramasuml.com/>
- [20] *Librería externa de ciudades y países* <https://www.npmjs.com/package/city-timezones>
- [21] *Imagen fuente del ciclo SCRUM* <https://www.colonet.de/blog/scrum-guide/scrum-im-selbststudium-teil-12-sprint-planning-planung-der-arbeit-des-sprints/>
- [22] *ODS* <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>

[23] Importancia del sector vitivinícola

<https://www.mapa.gob.es/es/agricultura/temas/producciones-agricolas/vitivinicultura/default.aspx#:~:text=Espa%C3%B1a%20cuenta%20con%20la%20mayor%20extensi%C3%B3n%20del%20cultivo,adem%C3%A1s%20de%20las%20empresas%20de%20suministros%20y%20maquinaria.>