



MÁSTER EN INGENIERÍA EN TECNOLOGÍAS DE INDUSTRIALES

TRABAJO FIN DE MÁSTER

Innovación en Gestión Sostenible:

Plataforma Digital para Reportes ESG basado en el estándar ESRS

Autor: Jorge Seco Martín

Director: Braulio Pareja Cano

Madrid

Junio de 2025

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

Innovación en Gestión Sostenible:

Plataforma Digital para Reportes ESG basado en el estándar ESRS

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2024/25 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido

tomada de otros documentos está debidamente referenciada.

Fdo.: Jorge Seco Martín

Fecha: 24 / 06 / 2025



Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Braulio Pareja Cano

Fecha: 24 / 06 / 2025



Agradecimientos

Quisiera expresar mi más profundo agradecimiento a mis padres por los sacrificios que hicieron para permitirme continuar con mis estudios.

A mi familia, gracias por su apoyo incondicional a lo largo de este camino.

También estoy sinceramente agradecido a Impact Bridge por haberme seleccionado para colaborar con ellos y por su apoyo durante el desarrollo de este proyecto.

Finalmente, deseo agradecer a mis profesores y al tutor del proyecto por haber hecho posible esta tesis.

INNOVACIÓN EN GESTIÓN SOSTENIBLE: PLATAFORMA DIGITAL PARA REPORTES ESG BASADO EN EL ESTÁNDAR ESRS

Autor: Seco Martín, Jorge.

Director: Pareja Cano, Braulio.

Entidad Colaboradora: Impact Bridge

RESUMEN DEL PROYECTO

Se presenta el desarrollo de una aplicación web para facilitar el cumplimiento del reporte no financiero de sostenibilidad (criterios ESG) en pequeñas y medianas empresas. La herramienta implementa una ubicación de una sola página (SPA) modular, alineado con los requisitos de los estándares ESRS y la directiva europea CSRD, que permite al usuario ingresar datos mediante formularios interactivos y obtener una vista preliminar de los resultados. En la demo actual se ha logrado cubrir las secciones clave (Introducción, Temas Clave, Medición, Resultados) con funciones de navegación, validación básica y presentación de información en tiempo real. La validación con usuarios (una pyme industrial, una startup de sostenibilidad y una empresa mediana) confirmó la utilidad del prototipo: se valoró su estructura clara por secciones, alineación normativa y capacidad de automatizar tareas, validando su potencial para PYMEs

Palabras claves: Desarrollo sostenible, Pequeñas y medianas empresas, Informes, ESG y CSRD

1. Introducción

En la última década, la sostenibilidad corporativa se ha convertido en una prioridad estratégica impulsada por nuevos requerimientos regulatorios. En particular, la UE ha aprobado la Directiva CSRD, que **extiende las exigencias** de reporte no financiero a un número mucho mayor de empresas, incluyendo por primera vez a muchas medianas. Junto a la CSRD surgen estándares detallados (ESRS) que introducen requisitos específicos como por ejemplo, la doble materialidad y métricas sectoriales, para la divulgación de datos ESG. Estas regulaciones, unidas a las mayores expectativas de mercado, requieren un nivel de flexibilidad y eficiencia en el manejo de datos que las herramientas convencionales de reporte no logran cubrir por completo. En general, las empresas afrontan el reto de adaptar sus procesos de recopilación y reporte ESG a exigencias más complejas, en un contexto donde el cumplimiento normativo puede traducirse además en ventaja competitiva (mayor transparencia ante clientes e inversores).

Por tanto, existe una clara **necesidad de nuevas herramientas** tecnológicas para agilizar el reporte ESG, especialmente orientadas a PYMEs. En las soluciones actuales ya sean plataformas genéricas de ESG o módulos integrados de ERP es común la entrada manual de datos, la falta de alineación con los nuevos estándares ESRS y la escasa adaptabilidad a organizaciones pequeñas. En este contexto surge el proyecto: se buscó desarrollar una **solución asequible y escalable** que automatice la recopilación y procesamiento de información ESG y garantice la alineación con la normativa vigente. Esta herramienta digital ayudará a las PYMEs no sólo a cumplir con sus obligaciones (reduciendo errores y trabajo manual), sino a integrar los criterios de sostenibilidad en su gestión diaria como un factor de eficiencia y valor añadido.

2. Definición del Proyecto

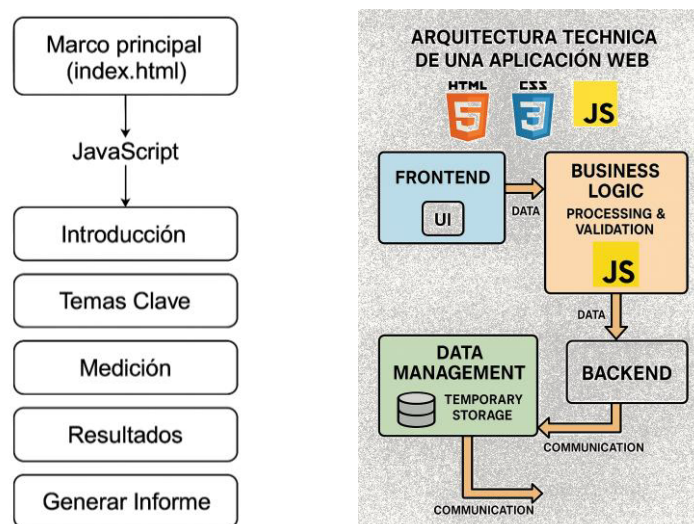
El objetivo principal del proyecto ha sido crear una **demo web interactiva** que sirva de prototipo para el reporte no financiero según ESRS/CSRD. La solución adoptada consiste en una aplicación web de página única (SPA) desarrollada con tecnologías web estándares (HTML5, CSS3 y JavaScript puro). Se ha diseñado una plataforma **ligera y de fácil despliegue**, accesible mediante cualquier navegador moderno, de modo que su adopción no suponga una barrera tecnológica incluso para organizaciones con recursos limitados. El enfoque está centrado en las **PYMEs**, dado que son las que enfrentan mayores dificultades con las soluciones comerciales disponibles. Por ello se optó por tecnologías abiertas y sin dependencias complejas; de hecho, se evitó el uso de frameworks externos (Bootstrap y React entre otros) para favorecer la simplicidad, la compatibilidad y el mantenimiento futuro.

La plataforma resultante está **alineada desde su concepción** con los marcos normativos ESG: incorpora explícitamente los requisitos de divulgación de ESRS y CSRD en su estructura de contenido. En la práctica, esto significa que el sistema guía al usuario para ingresar la información exactamente en los campos que exige la regulación y genera reportes enfocados a cubrir dichas exigencias. En este primer prototipo (demo), se ha implementado la mayoría de las secciones clave del informe de sostenibilidad, aunque ciertas funcionalidades avanzadas (como la generación automática del informe final) aún están planificadas para el desarrollo futuro. En términos concisos, el proyecto se propone mejorar la eficiencia del reporte ESG (automatizando tareas y minimizando errores) y crear una herramienta de bajo costo y alta flexibilidad, adaptada a las PYMEs y escalable a medida que estas crezcan.

3. Descripción del modelo/sistema/herramienta

La aplicación se organiza como una **SPA modular**: todas las vistas principales (Introducción, Temas Clave, Medición, Resultados, Generar Informe) residen en un único documento HTML y se muestran u ocultan dinámicamente sin recargar la página. La interfaz está dividida en secciones `<section>` referenciadas por el menú principal. Cuando el usuario selecciona un área (por ejemplo, «Temas Clave del Informe»), se invoca la función JavaScript `showSection(id)`, que asigna la clase CSS `active` a la sección correspondiente y elimina esa clase de las demás. Este mecanismo de manipulación del DOM permite que la navegación sea fluida y rápida: el contenido simplemente aparece en pantalla en el área principal, como se ilustra en la figura de arquitectura siguiente.

Figura 1: Diagrama conceptual de la arquitectura de la aplicación SPA.



Fuente: Elaboración propia

La navegación se complementa con mejoras de usabilidad: por ejemplo, se utiliza `scrollIntoView` para el desplazamiento suave hasta la sección activa, y se emplea `sessionStorage` para **preservar el estado de navegación** al recargar o trasladarse entre secciones. En el prototipo actual, el almacenamiento en sesión se usa principalmente para recordar qué sección estaba activa, evitando perder contexto al usuario (en futuras versiones se extenderá para guardar el progreso del llenado). Por motivos de privacidad y simplicidad, **los datos ingresados por el usuario no se almacenan permanentemente** en el navegador: permanecen en el DOM de la página y se eliminan al cerrar la sesión. Esto garantiza que no quede información sensible en el dispositivo, si bien implica que el usuario debe completar su informe en una sola sesión.

La entrada de datos se realiza mediante **formularios interactivos** basados en tablas. Cada estándar ESRS relevante se presenta como una tabla con columnas para el código, el requisito, el tipo de dato esperado y la respuesta del usuario. En la parte superior de cada formulario se muestra una **barra de progreso dinámica** que indica el avance en el llenado. Para permitir la edición directa, se habilitan las celdas de respuesta con el atributo HTML5 `contenteditable="true"`, convirtiéndolas en áreas editables al estilo de inputs de texto. De esta forma, el usuario puede hacer clic sobre la tabla y escribir o modificar sus respuestas in situ, sin necesidad de ventanas emergentes ni formularios adicionales. Esta aproximación mantiene la consistencia visual del formulario y ofrece una experiencia más natural e intuitiva que los controles de formulario convencionales.

La aplicación incluye además lógica de validación y cálculo en JavaScript: se comprueba que los campos marcados como obligatorios no queden vacíos y que los valores numéricos estén en rango razonable, alertando al usuario en caso contrario. Se implementaron scripts sencillos (por ejemplo, para el formulario inicial «¿Está tu empresa obligada?») que calculan automáticamente la aplicación de umbrales reglamentarios y ofrecen retroalimentación instantánea. El prototipo también está diseñado para incorporar en el futuro funciones de análisis avanzado de datos: por ejemplo, permitir proyecciones basadas en métricas ESG o integrar pequeños módulos de visualización de tendencias, para convertir los datos reportados en información estratégica (este componente analítico se mencionó en los objetivos del proyecto, pero queda pendiente de implementar en las siguientes fases). Por último, se dejó previsto que, al pulsar «Generar Informe», se combine todo lo ingresado en una vista preliminar. En esta primera versión, este enlace aún está inactivo como muestra la figura de navegación, pero se ha planificado el uso de la función nativa `window.print()` del navegador para producir un PDF de los resultados. De este modo, se mantendría la filosofía “light” del demo, sin necesidad de librerías externas, generando un informe imprimible fiel al contenido HTML mostrado.

4. Resultados

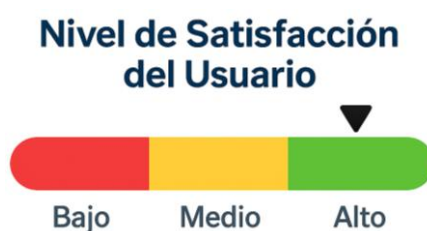
La demo web implementada demuestra la viabilidad de la propuesta: se logró un flujo de trabajo completo desde la introducción de datos hasta la visualización preliminar de resultados. Entre los **logros de la demo** destacan: la navegación rápida y coherente entre secciones de informe, la captura de datos mediante formularios claros (con indicadores de progreso) y la obtención de un borrador de informe que refleja la información ingresada. En las pruebas internas se comprobó que la plataforma efectivamente automatiza muchos pasos

del reporte ESG: por ejemplo, reduce drásticamente la carga manual al alinear de entrada los campos con los requisitos ESRS, evitando formularios innecesarios.

Además, se realizó una **validación piloto** de la plataforma con usuarios reales en tres contextos empresariales (una pequeña firma industrial, una startup de sostenibilidad y una empresa manufacturera mediana). Los tests confirmaron la aplicabilidad del prototipo a distintas realidades corporativas: organizaciones con poca experiencia en sostenibilidad como las pymes industriales obtuvieron una herramienta educativa para iniciarse en el reporte ESG, mientras que compañías con prácticas establecidas vieron un apoyo estructural para consolidar sus informes. En todos los casos, los evaluadores destacaron atributos comunes positivos del sistema: la estructura clara de secciones facilitó navegar del panorama general al detalle, la alineación con estándares reconocidos (CSRD/ESRS) dio confianza regulatoria, y la automatización de parte del proceso (pre-calcular cumplimientos, generar vistas formateadas) ahorró tiempo y proporcionó resultados consistentes. Estos beneficios clave: eficiencia operativa, guía normativa y estandarización, fueron valorados repetidamente por los usuarios piloto.

Los **feedbacks de prueba** también identificaron áreas de mejora. Sugirieron mejoras de usabilidad como ayudas contextuales (glosarios in-app, destacados de progreso) y recordatorios de secciones incompletas; propusieron funciones colaborativas (cuentas de usuario con roles, secciones bloqueables) para trabajo en equipo; y recomendaron permitir edición posterior al informe (plantillas flexibles y edición manual) para adaptar la salida final a cada caso. Estas sugerencias coinciden con la arquitectura modular seguida: las mejoras propuestas (autoguardado, multiusuario, edición avanzada) son factibles gracias al diseño desacoplado del código.

Figura 2: Ejemplo de indicador de satisfacción obtenido tras la validación del prototipo.



Fuente: Elaboración propia

En conjunto, los resultados actuales (capturados en la interfaz de resultados) muestran que la demo cumple con su objetivo de simplificar el reporte ESG. Como ejemplo ilustrativo, la Figura 2 representa un indicador agregado de satisfacción obtenido en las pruebas de usuario, el cual revela una valoración positiva de la plataforma (tendencia hacia las categorías verdes de alto cumplimiento y usabilidad). El prototipo alcanzó así su meta inicial: proporcionar a las PYMEs una **alternativa práctica y accesible** frente a los métodos tradicionales de reporte (hojas de cálculo básicas) o las plataformas empresariales complejas.

5. Conclusiones

La experiencia del desarrollo y validación del prototipo ha dejado varias lecciones aprendidas. En lo técnico, se confirma que una arquitectura web modular tipo SPA (HTML/CSS/JS) resulta suficiente para un prototipo de reporte ESG: ofrece transiciones rápidas, es fácilmente desplegable y permite crecer en funcionalidad sin reescrituras costosas. También se verificó la importancia de centrar el diseño en la **usabilidad para perfiles no técnicos**: tablas claras, progreso visible y validaciones inmediatas aumentan la intuitividad de la herramienta. A nivel de contenido, alinear la plataforma con el marco ESRS/CSRD desde el inicio simplifica enormemente el trabajo del usuario, confirmando que esta estandarización normativa es clave para generar reportes fiables y comparables.

En términos de utilidad, la validación demuestra que el prototipo es **viable y valioso para su público objetivo**. En palabras de los pilotos, la herramienta es una “aliada estratégica” para las PYMEs, no sólo para cumplir la normativa sino para potenciar su competitividad en la era ESG. Este resultado concuerda con estudios de la industria que destacan que la tecnología de información es determinante para generar reportes más eficaces. En definitiva, la demo aporta transparencia y estructura al proceso de reporte, reduciendo errores y esfuerzo manual en un sector históricamente desatendido.

Próximos pasos de desarrollo. Sobre la base de las lecciones y feedback obtenidos, el camino futuro incluye la implementación de mejoras claves: por ejemplo, autoguardado y notificaciones para evitar pérdida de datos, interfaces colaborativas con cuentas multiusuario, y un motor de reporte avanzado con plantillas adaptables. En particular, se prevé completar la generación de informes en formatos PDF/XBRL (usando la impresión nativa o librerías especializadas), así como reforzar la validación de entradas y la seguridad de los datos. Gracias al diseño modular y escalable, muchas de estas ampliaciones pueden incorporarse de forma incremental. En suma, el trabajo futuro orientará el paso del prototipo

hacia un producto plenamente implementable en entornos reales, con el objetivo de que más PYMEs puedan afrontar los desafíos del reporting ESG de forma eficaz y alineada con las mejores prácticas internacionales

6. Referencias

1. EFRAG. (2023). ESRS Exposure Drafts – European Sustainability Reporting Standards. European Financial Reporting Advisory Group. Recuperado de <https://www.efrag.org>
2. Comisión Europea. (2022). Corporate Sustainability Reporting Directive (CSRD). Diario Oficial de la Unión Europea. Recuperado de <https://eur-lex.europa.eu>
3. García, M., & López, A. (2021). Desarrollo de plataformas digitales para el reporte no financiero: una revisión del estado del arte. Revista Ingeniería y Sostenibilidad, 14(2), 45–62. <https://ojs.unipamplona.edu.co/index.php/iss>

Índice de la memoria

Capítulo 1. Introducción	7
1.1 Definición de Objetivos.....	7
1.2 Oportunidad de Mercado.....	9
1.3 Enfoque del proyecto y propuesta de solución.....	10
1.4 Metodología.....	12
Capítulo 2. Contexto regulatorio y estratégico del ESG	14
2.1 Tendencias normativas y estándares comunes	15
2.2 Criterios y fechas.....	15
2.2.1 Grandes empresas	15
2.2.2 PYMEs orientadas al mercado de capitales.....	16
2.2.3 Empresas no europeas.....	16
2.2.4 Formulario interactivo de obligaciones	16
2.3 Relevancia estratégica y presión de stakeholders.....	18
2.4 Barreras y desafíos en las PYMEs.....	18
2.5 Digitalización del reporte ESG: solución propuesta	19
Capítulo 3. Tecnologías Seleccionadas y justificación técnica.....	21
3.1 Tecnologías empleadas.....	21
Capítulo 4. Estructura modular y arquitectura de la aplicación.....	28
4.1 Organización modular en archivos HTML.....	28
4.2 Navegación controlada por JavaScript entre módulos	31
4.3 Lógica estructural de la interfaz y flujo de navegación.....	33
4.4 Escalabilidad, mantenimiento y adaptación futura de la arquitectura	38
Capítulo 5. Navegación e Interactividad con JavaScript.....	42
5.1 Introducción.....	42
5.2 Estructura de Navegación tipo SPA	43
5.3 Implementación de la Función showSection(id)	46
5.4 Desplazamiento Suave y Manejo de Eventos del DOM.....	48
5.5 Persistencia del Estado de Navegación con SessionStorage	51
5.6 Mejora de la experiencia visual con animaciones	54

5.7 Conclusiones de la Implementación de Navegación	54
Capítulo 6. Entrada de Datos y Diseño de Formularios	56
6.1 Formulario en la sección de Introducción	56
6.2 Uso de tablas como interfaz principal para la entrada de datos.....	57
6.3 Implementación de campos editables mediante contenteditable	58
6.4 Lógica de gestión de los datos introducidos (DOM, almacenamiento temporal, eventos) .	59
6.5 Interfaz orientada a la usabilidad: claridad visual, navegación y organización	61
6.6 Integración de una barra de progreso en los formularios	65
6.7 Propuestas de mejora.....	67
Capítulo 7. Visualización de Resultados y Generación del Informe	69
7.1 Visualización de resultados	69
7.2 Vista previa del informe	71
7.3 Generación del informe (concepto de desarrollo futuro).....	73
7.4 Privacidad y seguridad de los datos.....	75
Capítulo 8. Adaptabilidad a PYMEs e Implementación Ligera.....	78
8.1 Diseño de la aplicación centrado en las PYMEs	79
8.2 Implementación ligera y despliegue simplificado	80
8.3 Alineación con los requisitos de la Directiva CSRD.....	82
8.4 Escalabilidad técnica y mejoras futuras propuestas	85
8.5 Perspectiva estratégica: justificación de la solución para PYMEs y barreras de entrada en el mercado.....	88
Capítulo 9. Validación del Prototipo con Usuarios.....	91
9.1 Validación del Prototipo en Entornos Empresariales	91
9.2 Validación en una pequeña empresa industrial (Grupo Industrial García).....	92
9.3 Validación en una startup de sostenibilidad	96
9.4 Empresa manufacturera mediana	102
9.5 Síntesis de la validación y recomendaciones finales	104
Capítulo 10. Despliegue Técnico Sencillo	107
10.1 Hoja de ruta de despliegue inicial	107
10.2 Posible migración a dominio propio y servidor	111
10.3 Evolución hacia un backend (Node.js, PHP, base de datos)	112

10.4 Escalabilidad y mantenimiento.....	115
10.5 Recomendaciones para un despliegue profesional futuro	117
Capítulo 11. Conclusiones y Trabajos Futuros.....	121
11.1 Resumen de logros del proyecto	121
11.2 Lecciones aprendidas	121
11.3 Implementación futura	123
11.4 Reflexión final.....	123
Capítulo 12. Bibliografía.....	125

Índice de tablas

Tabla 1. Comparativa entre requisitos de la Directiva CSRD y la funcionalidad de la aplicación desarrollada, indicando mejoras futuras para reforzar la alineación.....	84
Tabla 2: Líneas de mejora	123

Índice de figuras

Figura 1: Diagrama conceptual de la arquitectura de la aplicación SPA.	7
Figura 2: Ejemplo de indicador de satisfacción obtenido tras la validación del prototipo....	9
Figura 3: Sección de “¿Está tu empresa obligada?” en la página de Inicio.....	17
Figura 4: Organización modular de la página web.....	30
Figura 5: Etapa 1 de flujo típico	34
Figura 6: Etapa 2 de flujo típico	35
Figura 7: Etapa 3 de flujo típico	36
Figura 8: Etapa 4 de flujo típico	37
Figura 9: Etapa 5 de flujo típico	37
Figura 10: Ejemplo de estructura HTML con secciones diferenciadas y clase active para navegación entre bloques.....	43
Figura 11: Fragmento de código con enlaces a secciones mediante onclick y showSection	44
Figura 12: Interfaz de la plataforma mostrando la sección "Temas Clave del Informe"	45
Figura 13: Código de la función showSection para la navegación entre secciones	46
Figura 14: Reglas CSS para mostrar y ocultar secciones mediante la clase active	47
Figura 15: Código para desplazamiento suave y detección de hash en el evento	48
Figura 16: Ejemplo de invocación directa para mejorar la navegación entre secciones	49
Figura 17: Botón “Siguiente” con navegación secuencial entre secciones mediante.....	50
Figura 18: Persistencia de la sección activa usando sessionStorage al recargar o regresar	52
Figura 19: Código para retener la navegación entre secciones tras volver a la página	52
Figura 20: Interfaz de entrada de datos basada en una tabla para el estándar ESRS E1	57

Figura 21: Fragmento para mantener la sección activa tras volver a la página.....	60
Figura 22: Pantalla de Temas Clave del Informe	62
Figura 23: Pantalla de que ilustra el menú superior de navegación.	63
Figura 24: Botones de navegación del menú Temas clave.....	63
Figura 25: Función utilizada para la navegación entre formularios.	64
Figura 26: Definición de estilos CSS para la barra de progreso de los formularios.....	65
Figura 27: Formulario con barra de progreso dinámica y visualización de cumplimiento .	66
Figura 28: Ejemplo de visualización de resultados para el estándar Cambio Climático.....	70
Figura 29: Botón descargar informe en página resultados	73
Figura 30: Perfiles de usuarios reales en la prueba del prototipo	92
Figura 31: Interfaz con datos personales de los puntos del estándar ESRS E1	94
Figura 32: Pantalla principal mostrando la navegación por secciones y la lista de “Temas Clave del Informe”	98

Capítulo 1. INTRODUCCIÓN

En la última década, la sostenibilidad empresarial ha dejado de ser un valor añadido para convertirse en una prioridad estratégica. La creciente presión regulatoria y las demandas de transparencia por parte de los grupos de interés (stakeholders) están redefiniendo la forma en que las empresas gestionan sus **factores ESG** (criterios ambientales, sociales y de gobernanza). En particular, la Unión Europea ha impulsado nuevas normativas como la **Directiva CSRD** (Corporate Sustainability Reporting Directive), que extiende las exigencias de reporte de información no financiera a un número mucho mayor de empresas, incluyendo por primera vez a muchas medianas empresas. Vinculados a esta directiva surgen estándares detallados como el **ESRS** (European Sustainability Reporting Standards), los cuales establecen requisitos específicos, por ejemplo, la consideración de la doble materialidad y métricas sectoriales obligatorias para la divulgación de datos ESG.

Este nuevo entorno normativo, junto con unas expectativas de mercado cada vez más elevadas en materia de sostenibilidad, requiere un nivel de flexibilidad y eficiencia en el manejo de datos ESG que las herramientas convencionales de reporte no logran cubrir por completo. Las empresas se enfrentan así al reto de adaptar sus procesos de recopilación y reporte de información no financiera a unas exigencias más estrictas y complejas, en un contexto donde cumplir con la sostenibilidad no solo evita sanciones, sino que también **puede traducirse en una ventaja competitiva** al satisfacer las expectativas de clientes e inversores.

1.1 DEFINICIÓN DE OBJETIVOS

El **objetivo principal** es **desarrollar una aplicación web modular de una sola página (SPA) que facilite la elaboración de reportes de sostenibilidad (criterios ESG) para pequeñas y medianas empresas, alineada con el estándar ESRS y la normativa CSRD.**

A continuación, se expone su desarrollo y se contextualiza en relación con las necesidades identificadas en el entorno empresarial actual.

En el desarrollo de este objetivo principal se han tenido en cuenta las limitaciones de las herramientas existentes y los requisitos normativos emergentes. Concretamente, se busca ofrecer una plataforma accesible y económica que permita a las PYMEs incorporar de forma natural los criterios ESG en sus procesos operativos diarios, más allá del mero cumplimiento legal. Para ello, la aplicación diseñada incluye formularios interactivos para la entrada de datos ESG, módulos de validación y análisis, y una interfaz de usuario amigable que garantiza la correcta adaptación a los nuevos estándares de reporte no financiero. De este modo, el software no solo cumple con las exigencias legales, sino que también aporta valor añadido al transformar los datos de sostenibilidad en información estratégica para la toma de decisiones.

Además del objetivo principal, se plantean los siguientes **objetivos secundarios**, que desglosan y concretan las metas a corto plazo del proyecto:

- Analizar el marco normativo y regulatorio: Investigar los estándares ESRS y la directiva CSRD, así como la relevancia de los Objetivos de Desarrollo Sostenible (ODS) para las empresas, de manera que se obtenga un corpus teórico robusto que sustente el diseño de la herramienta.
- Identificar los requerimientos y necesidades de las PYMEs: Realizar un estudio de requisitos que incluya la recopilación de datos de potenciales usuarios (empresas industriales, startups de sostenibilidad, etc.), con el fin de determinar las funcionalidades y criterios clave que debe tener la aplicación para ser realmente útil y viable en el mercado.
- Diseñar la arquitectura del sistema: Plantear una solución técnica modular y escalable, seleccionando las tecnologías y frameworks adecuados (por ejemplo, frameworks web modernos, lenguajes de programación y bases de datos) para implementar la plataforma SPA de forma eficiente.

- Implementar un prototipo funcional: Desarrollar las secciones esenciales de la aplicación (Introducción, Temas Clave, Medición, Resultados, etc.), incluyendo la creación de formularios dinámicos, validaciones automáticas y visualizaciones de información en tiempo real, siguiendo las mejores prácticas de desarrollo de software.
- Realizar pruebas y validación de usuario: Evaluar el prototipo mediante pruebas técnicas (unitarias, integración) y validación con usuarios finales (PYMES reales y expertos del sector), recogiendo feedback para corregir errores, ajustar la interfaz y garantizar que la herramienta cumple con los objetivos planteados.
- Documentar y difundir la solución: Elaborar la documentación técnica y académica del proyecto, incluyendo esta memoria del TFM, y preparar materiales de presentación (informe, demostración) que expliquen cómo la plataforma ayuda a las PYMES a lograr los objetivos ESG y normativos.

Estos objetivos secundarios complementan el objetivo principal y ofrecen una guía detallada de las acciones a realizar. En la conclusión del trabajo se volverá a hacer referencia a estos objetivos para evaluar en qué medida la plataforma desarrollada y el proceso seguido han contribuido a alcanzarlos. Se explicará cómo cada objetivo secundario ha sido cumplido y cómo ello refuerza la consecución del objetivo principal, evidenciando la aportación del proyecto en el contexto de la gestión sostenible en PYMES.

1.2 OPORTUNIDAD DE MERCADO

Por otro lado, existen **soluciones integradas en sistemas ERP** (Enterprise Resource Planning) orientadas a la sostenibilidad, como **SAP Sustainability Control Tower** o **Microsoft Sustainability Manager**, que incorporan módulos ESG dentro de la gestión empresarial integral. Estas soluciones permiten cierto grado de **automatización** en la recopilación de datos (aprovechando la integración con otras fuentes internas) y pueden generar informes alineados con estándares internacionales. No obstante, **presentan sus propias limitaciones**: su implementación suele ser costosa y está principalmente dirigida a grandes corporaciones, requiriendo una integración profunda en los procesos y sistemas

existentes. Esto las vuelve poco adecuadas para **pequeñas y medianas empresas (PYMEs)**, que generalmente carecen de la infraestructura o los recursos para desplegar herramientas de tal envergadura. Incluso en organizaciones mayores, la rigidez de estas soluciones integradas puede implicar menor flexibilidad para adaptarse rápidamente a cambios normativos o a necesidades específicas fuera del alcance predeterminado del software.

Sintetizando la idea, tras analizar el panorama actual, se identifican **varios posibilidades tecnológicas de entrada**: (1) con la entrada en vigor de estándares como ESRS, muchas empresas carecen de herramientas alineadas que automaticen el cumplimiento de estos requerimientos; (2) las plataformas disponibles no garantizan una automatización completa ni la verificación de la calidad de todos los datos ESG recopilados, teniendo que recurrir en parte a procesos manuales; (3) pocas soluciones están pensadas para integrarse fácilmente con sistemas existentes de las empresas y **escalar** en funcionalidad conforme crecen las necesidades; y (4) las funcionalidades de análisis suelen limitarse a la retrospectiva, sin brindar apoyo para proyecciones o análisis avanzados que ayuden en la **toma de decisiones estratégicas** basadas en los datos de sostenibilidad. Todo ello pone de manifiesto la necesidad de una herramienta más **personalizada, accesible y adaptable** a este nuevo contexto.

1.3 ENFOQUE DEL PROYECTO Y PROPUESTA DE SOLUCIÓN

Como respuesta a las necesidades anteriores, se ha enfocado en el **desarrollo de una aplicación web en HTML** para facilitar el **reporte no financiero** de las empresas conforme a los nuevos estándares ESG. En esencia, se ha diseñado una plataforma web ligera, de fácil acceso a través de una página web, utilizando **tecnologías abiertas, sencillas y fácilmente desplegables**, de modo que su adopción no suponga una barrera incluso para organizaciones con recursos informáticos limitados. El **enfoque está centrado en las PYMEs**, dado que son estas las que más dificultades encuentran con las soluciones comerciales existentes: la herramienta propuesta busca proporcionarles una alternativa económica y flexible para cumplir con sus obligaciones de sostenibilidad. A diferencia de

las plataformas genéricas o las suites integradas mencionadas anteriormente, la solución desarrollada en este proyecto se caracteriza por estar **alineada desde su concepción con el marco ESRS** y la normativa derivada de la CSRD, incorporando en su estructura los requisitos de divulgación específicos que estos establecen. De este modo, una empresa usuaria de la plataforma podrá saber que los datos que introduce y los informes que genera están directamente enfocados a cubrir lo que la regulación europea exige, reduciendo el riesgo de incumplimientos por desajuste con el estándar.

Asimismo, el software creado y sus futuras mejoras aportan una serie de **mejoras diferenciales** frente a las herramientas actuales, orientadas a solventar las carencias detectadas en estas. En particular, la plataforma ofrece:

- **Escalabilidad y flexibilidad** para adaptarse a distintas realidades empresariales. La solución está pensada con una arquitectura modular y parametrizable, que permite su uso tanto en pequeñas empresas como en corporaciones de mayor tamaño, adecuándose a diferentes sectores. Esto implica que la herramienta puede crecer en funcionalidad o capacidad a medida que la organización lo requiera, sin necesidad de reimplementaciones costosas.
- **Herramientas de análisis avanzado** que transforman los datos ESG recopilados en **información estratégica**. Más allá de simplemente cumplir con el reporte, la plataforma incorpora módulos de visualización y análisis de datos que ayudan a la empresa a interpretar su desempeño en sostenibilidad. Incluso se plantean funcionalidades de análisis predictivo o simulación, de forma que los responsables puedan proyectar escenarios futuros basados en métricas ESG y apoyar la toma de decisiones informadas en materia de sostenibilidad.
- **Automatización avanzada** en la recopilación y procesamiento de datos ESG, reduciendo la carga operativa manual y minimizando errores. Mediante esta automatización (por ejemplo, integración de scripts o pequeños programas para importar datos desde fuentes internas), se acelera la generación de informes y se asegura una mayor precisión en la información reportada.

Con esta propuesta, el proyecto no solo atiende al **cumplimiento normativo** como imperativo, sino que también **promueve la integración de la sostenibilidad en la gestión empresarial diaria**. En otras palabras, la herramienta está diseñada para que las empresas usuarias puedan **incorporar los criterios ESG en su operativa** de manera natural y continua, más allá de la obligación de reportar periódicamente. De este modo, se espera que su uso facilite **la identificación de áreas de mejora** en el desempeño ambiental, social y de gobierno corporativo de cada organización, así como **la elaboración de informes claros, transparentes y verificables** que fortalezcan la confianza de los stakeholders en la información divulgada. Adicionalmente, al alinear la plataforma con marcos internacionales de sostenibilidad, **se incentiva la contribución a los Objetivos de Desarrollo Sostenible (ODS)**, ayudando a las empresas a estructurar y amplificar su impacto social y ambiental positivo en consonancia con dichos objetivos globales.

Por último, cabe destacar que el presente proyecto se ha desarrollado en **colaboración con la empresa Impact Bridge SL**, entidad especializada en inversiones sostenibles y proyectos de impacto social alineados con los ODS de la ONU. Esta colaboración estratégica ha aportado un **enfoque práctico y orientado al mercado** durante todo el desarrollo del software. Gracias a la participación de Impact Bridge, se han podido contrastar las funcionalidades y enfoques de la herramienta con casos y necesidades reales del sector, lo que asegura que el resultado final sea **relevante y viable en un contexto empresarial real**. En suma, la alianza con una firma experta en sostenibilidad ha garantizado que la solución propuesta esté **alineada con las expectativas de los inversores y de las empresas sostenibles**, dotando al proyecto de un valor añadido y una validación en cuanto a su utilidad en el mundo profesional.

1.4 METODOLOGÍA

El desarrollo de este se ha estructurado en varias fases claramente diferenciadas para garantizar la coherencia técnica, la viabilidad normativa y la orientación práctica de la solución propuesta.

- **Fase 1:** Análisis del contexto y marco normativo. Se revisaron directivas, estándares ESRS y literatura técnica relacionada con el reporte ESG para pequeñas y medianas empresas. Se consultaron fuentes como EFRAG, Comisión Europea y publicaciones académicas especializadas.
- **Fase 2:** Definición de requisitos funcionales. A partir del análisis normativo y del estudio de mercado, se definieron los requerimientos de la herramienta: estructura modular, formularios interactivos, validación básica y facilidad de despliegue.
- **Fase 3:** Diseño técnico de la arquitectura. Se diseñó una arquitectura SPA (Single Page Application) basada en HTML5, CSS3 y JavaScript, priorizando la simplicidad, la independencia de frameworks externos y la usabilidad para usuarios no técnicos.
- **Fase 4:** Desarrollo y prototipado. Se implementó un prototipo funcional que cubre las etapas de introducción, temas clave y medición. La lógica de navegación, edición y validación se codificó en JavaScript puro.
- **Fase 5:** Validación con usuarios. Se realizaron pruebas piloto con tres perfiles empresariales (pyme industrial, startup de sostenibilidad y empresa manufacturera mediana) para comprobar la alineación normativa y la experiencia de uso.
- **Fase 6:** Documentación y conclusiones. Finalmente, se elaboró esta memoria técnica, complementada con figuras y resultados, para evidenciar la contribución del proyecto a la mejora del reporte ESG en PYMEs.

En conclusión, este capítulo introductorio ha presentado la **motivación y el contexto** que dan origen al proyecto, evidenciando la necesidad de nuevas herramientas de reporte ESG ante la evolución del marco normativo (ESRS, CSRD) y las limitaciones de las soluciones existentes, especialmente para las PYMEs. Asimismo, se ha descrito el **enfoque adoptado** y las características clave de la **plataforma web desarrollada**, enfatizando su valor diferencial frente a las alternativas comerciales y su orientación a facilitar la sostenibilidad como eje estratégico empresarial. En los siguientes capítulos se profundizará en los **objetivos concretos**, el **diseño e implementación** de la solución propuesta y la **evaluación de sus resultados**, asentando las bases técnicas y metodológicas que sustentan este trabajo.

Capítulo 2. CONTEXTO REGULATORIO Y ESTRATÉGICO DEL ESG

En los últimos años el enfoque ESG ha cobrado una importancia central tanto en la regulación como en la estrategia corporativa. A nivel europeo, la **Directiva CSRD** reemplaza a la antigua directiva de información no financiera, ampliando el volumen y detalle de los reportes. La CSRD exige un **doble proceso de materialidad**, evaluar tanto los riesgos ESG del negocio como su impacto social y ambiental y extiende las obligaciones a un número mucho mayor de empresas, incluyendo por primera vez muchas medianas. Además, se están desarrollando estándares comunes (ESRS) bajo la tutela del EFRAG, inspirados en marcos internacionales como el GRI. En conjunto, el nuevo marco normativo obliga a las empresas a integrar la sostenibilidad en sus informes oficiales, mejorando la **transparencia y comparabilidad** de la información ESG. Como señalan expertos, una mayor claridad y calidad del reporting ESG también se traduce en mayor capacidad de atraer inversión y financiación. [Carrillo, A. J. (2024, January 16)].

Por otra parte, la **presión social y de mercado** empuja a las empresas hacia la sostenibilidad. Existe una creciente sensibilización de consumidores, inversores y ciudadanos hacia las prácticas ambientales y sociales de las empresas. La opinión pública cada vez exige más **legitimidad social**: las empresas deben demostrar que actúan de manera ética y responsable. De hecho, apuntan que: integrar la sostenibilidad puede fortalecer la imagen pública, aumentar la lealtad de clientes y empleados, y abrir acceso a nuevos mercados o financiación preferente. En este sentido, muchas compañías ven el ESG no solo como un requisito legal sino como una **oportunidad competitiva** a largo plazo. [Asociación interamericana de contabilidad. (2024, October 5)].

2.1 TENDENCIAS NORMATIVAS Y ESTÁNDARES COMUNES

La Unión Europea lidera diversas iniciativas regulatorias que refuerzan el foco ESG. Junto con la CSRD, destacan el **Reglamento de Taxonomía** y la **Debida Diligencia Corporativa** (CSDDD). Por ejemplo, la Taxonomía UE (Regl. 2020/852) define qué actividades pueden considerarse ambientalmente sostenibles, conectándose con los reportes ESG. Además, el CSRD en desarrollo exige que las grandes empresas establezcan planes de sostenibilidad alineados con el Acuerdo de París. En paralelo, organismos internacionales como la IFRS Foundation están elaborando estándares globales (ISSB) para facilitar la comparabilidad.

Estas tendencias normativas reflejan la necesidad de homogeneizar la información no financiera: la CSRD (y sus borradores ESRS) busca estandarizar las divulgaciones, mejorando la calidad de los reportes y permitiendo comparaciones entre empresas e industrias. En resumen, el marco regulatorio actual exige a las empresas un esfuerzo creciente en reporte ESG: mayores volúmenes de datos, indicadores precisos (incluso sectoriales) y aseguramiento de la información, lo que supera ampliamente los requisitos previos bajo la NFRD.

2.2 CRITERIOS Y FECHAS

A continuación, se detalla a quién afecta el **CSRD**.

2.2.1 GRANDES EMPRESAS

Con la reforma aprobada en 2024, la CSRD se aplicará a partir de 2025 a todas aquellas empresas que cumplan dos de los tres criterios siguientes:

- Más de **1.000 empleados** (anteriormente el umbral era 250).
- Un balance total superior a **25 millones €**.
- Un volumen neto de negocios superior a **50 millones €**.

Estas empresas deberán presentar su primer informe bajo CSRD en **2026** sobre el ejercicio 2025. La modificación del umbral responde al objetivo de centrar los esfuerzos normativos en compañías con mayor capacidad de reporte y evitar una carga excesiva para empresas medianas.

2.2.2 PYMES ORIENTADAS AL MERCADO DE CAPITAL

Las **pequeñas y medianas empresas cotizadas**, excluyendo microempresas, estarán obligadas a reportar información bajo CSRD a partir del **1 de enero de 2026**, con su primer informe en **2027**. No obstante, pueden acogerse a un periodo de gracia que les permite **excluirse voluntariamente hasta 2028**.

Según la Directiva 2013/34/UE, una empresa se considera pequeña si cumple al menos dos de los siguientes criterios:

- Hasta **10 empleados**.
- Balance inferior a **350.000 €**.
- Volumen de negocios neto inferior a **700.000 €**.

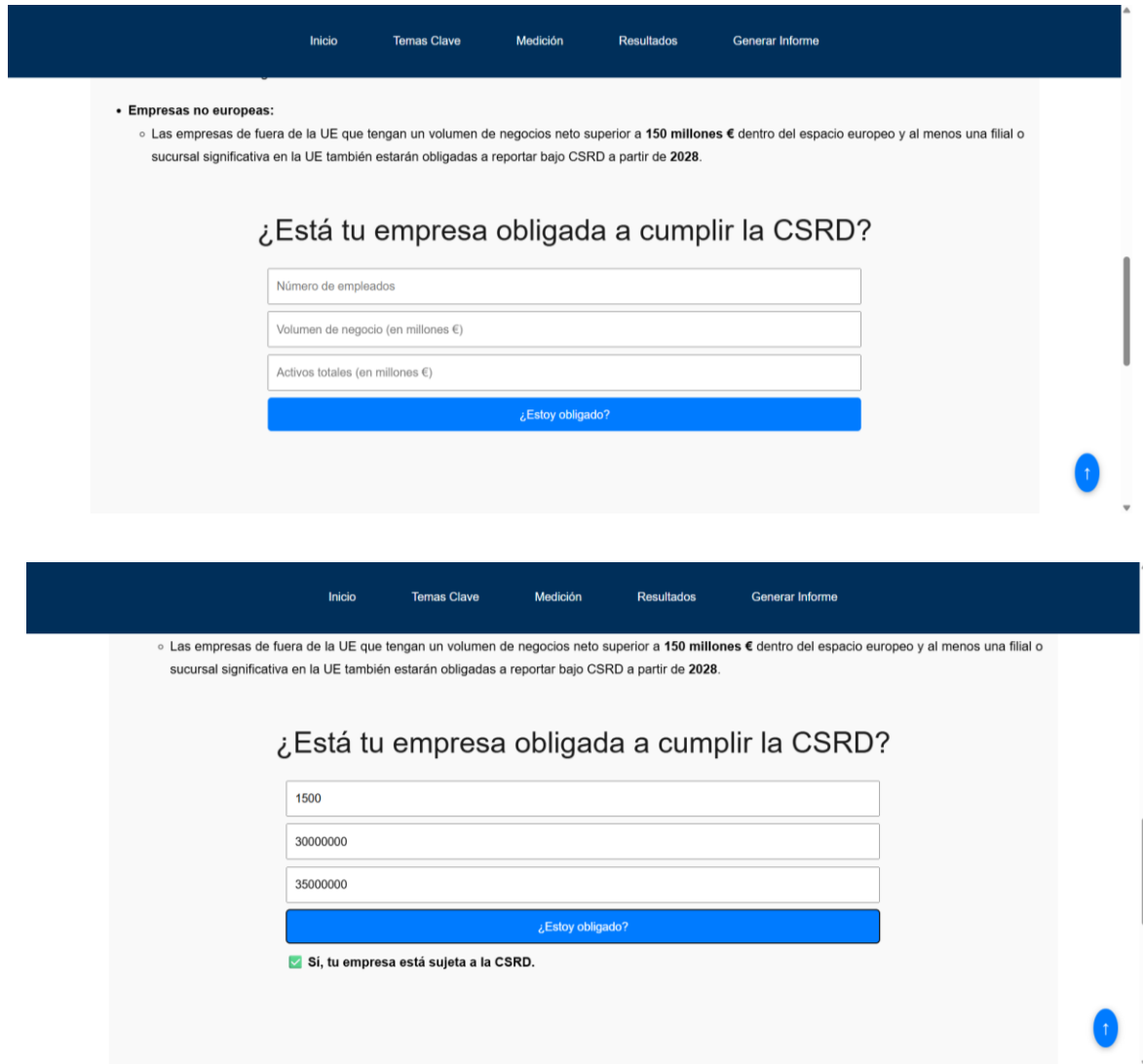
2.2.3 EMPRESAS NO EUROPEAS

Las empresas de fuera de la UE que tengan un volumen de negocios neto superior a **150 millones €** dentro del espacio europeo y al menos una filial o sucursal significativa en la UE también estarán obligadas a reportar bajo CSRD a partir de **2028**.

2.2.4 FORMULARIO INTERACTIVO DE OBLIGACIONES

La web incorpora un cuestionario en la página de inicio (“¿Está tu empresa obligada a cumplir la CSRD?”) con campos de número de empleados, volumen y activos, junto a un botón “**¿Estoy obligado?**”. Este formulario interactivo, es un cálculo implementado en JavaScript que evalúa automáticamente si la empresa supera los umbrales definidos.

Figura 3: Sección de “¿Está tu empresa obligada?” en la página de Inicio



The figure consists of two screenshots of a web application interface. Both screenshots show a dark blue navigation bar at the top with the following links: Inicio, Temas Clave, Medición, Resultados, and Generar Informe.

Top Screenshot: Below the navigation bar, there is a section titled "Empresas no europeas:" with a sub-point: "Las empresas de fuera de la UE que tengan un volumen de negocios neto superior a 150 millones € dentro del espacio europeo y al menos una filial o sucursal significativa en la UE también estarán obligadas a reportar bajo CSRD a partir de 2028." Below this text is a form titled "¿Está tu empresa obligada a cumplir la CSRD?". The form contains three input fields: "Número de empleados", "Volumen de negocio (en millones €)", and "Activos totales (en millones €)". At the bottom of the form is a blue button labeled "¿Estoy obligado?".

Bottom Screenshot: This screenshot shows the same form after it has been filled out. The input fields now contain the values: "1500", "30000000", and "35000000". The blue button "¿Estoy obligado?" is still present. Below the button, there is a green checkmark icon followed by the text: "Si, tu empresa está sujeta a la CSRD."

Fuente: Elaboración propia

Pantallazo de la sección de “¿Está tu empresa obligada?” (pág. inicio), mostrando los campos de entrada y el botón. La primera imagen antes de ser completada, y la segunda posterior.

2.3 *RELEVANCIA ESTRATÉGICA Y PRESIÓN DE STAKEHOLDERS*

Las demandas regulatorias se suman a exigencias de **mercado y sociedad**. Clientes, inversores y otros grupos de interés piden cada vez más información sobre sostenibilidad; muchas grandes empresas la exigen a sus proveedores y socios de la cadena de valor. Por tanto, incorporar criterios ESG ya no es solo un imperativo ético, sino un factor clave de competitividad. Estudios recientes muestran que la sostenibilidad es una de las principales preocupaciones de las PYMEs, precisamente porque afecta la confianza de clientes, inversores y ciudadanía. Como resultado, las empresas que mejoran su reporte ESG pueden diferenciarse positivamente: según informes, el **80% de las PYMEs** considera beneficioso esforzarse en sostenibilidad, sobre todo por la mejora de reputación que conlleva. En definitiva, la transparencia en criterios ambientales, sociales y de gobernanza se asocia a ventajas competitivas (imagen de marca, fidelización de talento, acceso a mercados internacionales, eficiencia operativa). [elEconomista.es. (2024, July 1)].

2.4 *BARRERAS Y DESAFÍOS EN LAS PYMES*

A pesar de la importancia reconocida, muchas PYMEs enfrentan **obstáculos significativos** para adoptar el enfoque ESG. Su tamaño limitado suele implicar **recursos humanos y financieros escasos**, lo que dificulta dedicar personal especializado a sostenibilidad. Asimismo, la falta de conocimiento técnico o de guías sencillas sobre ESG amplía la percepción de que estas exigencias son “sobrerregulación” ajena a su realidad. En concreto, estudios de organizaciones empresariales señalan que más del 80% de PYMEs identifica barreras como costes elevados y complejidad de requerimientos. Estas dificultades se agravan porque las PYMEs suelen depender de herramientas muy básicas (por ejemplo, hojas de cálculo genéricas) o de plataformas empresariales diseñadas para grandes corporaciones, que resultan costosas, rígidas o poco intuitivas para empresas pequeñas. [elEconomista.es. (2024, July 1)].

Frente a esto, la instauración del reporting ESG en PYMEs exige soluciones proporcionadas a su escala. En el capítulo anterior se identificó que las herramientas

comerciales existentes no satisfacen completamente estas necesidades: requieren mucho trabajo manual, no están adaptadas a los nuevos estándares ESRS y, en general, su complejidad supera las capacidades técnicas de las PYMEs. Además, la ausencia de funciones guía (por ejemplo, asesores virtuales o matrices de materialidad simplificada) hace que el proceso sea arduo y poco accesible. Esto crea un vacío tecnológico: las PYMEs necesitan soluciones más sencillas, económicas y específicas.

2.5 DIGITALIZACIÓN DEL REPORTE ESG: SOLUCIÓN PROPUESTA

En este contexto, las **herramientas digitales especializadas** juegan un papel esencial para mejorar el cumplimiento, la transparencia y la eficiencia del reporting ESG. Organismos como EFRAG han empezado a ofrecer recursos digitales gratuitos (plantillas Excel interactivas, taxonomías XBRL, convertidores automáticos) para facilitar la generación de informes de sostenibilidad estandarizados, una de las fuentes principales usadas para cumplir con los requerimientos. Por ejemplo, la «plantilla digital VSME» del estándar voluntario para PYMEs permite recopilar datos con validaciones automáticas y cálculos preconfigurados, y el uso de XBRL asegura intercambiar la información en formato legible por máquina. Estas iniciativas buscan eliminar la duplicidad de esfuerzos y reducir la carga administrativa, un problema real hoy: muchas PYMEs deben completar múltiples cuestionarios ESG distintos con criterios no estandarizados, lo cual genera costes innecesarios. [Amec Positive Industry. (2025, May 29)].

La **herramienta web desarrollada** responde precisamente a esas necesidades. Se trata de una plataforma HTML ligera que organiza los formularios por estándares ESRS, verificando la integridad de los datos e incluyendo explicaciones integradas de cada indicador. Gracias a ello, un usuario sin perfil técnico puede completar un informe ESG siguiendo un flujo guiado, casi como si fuera un “asesor virtual”. El software emplea tecnología abierta, lo que permite ofrecerlo a bajo costo (con una versión inicial gratuita para dar a conocer el producto) y reducir así la barrera económica para las PYMEs. Además, no requiere instalaciones complejas ni personal de TI dedicado, cumpliendo con la realidad de muchas empresas pequeñas.

Por otra parte, la herramienta facilita la proporcionalidad y relevancia del reporte: al poder configurarse según el tamaño y sector de la empresa, permite centrarse solo en los indicadores esenciales de ESG. Esto está alineado con la tendencia hacia estándares simplificados para PYMEs (por ejemplo, el VSME) y con las previsiones regulatorias que contemplan periodos de transición (opt-out) para pequeños sujetos obligados. Finalmente, el proceso interactivo de “aprender haciendo” que incorpora la aplicación ayuda a las PYMEs a ganar conocimiento interno sobre sus impactos y riesgos ESG, algo señalado como valioso en la literatura.

En síntesis, tal como se concluye en capítulos anteriores, la **digitalización del reporte ESG** mediante esta herramienta aporta mayor transparencia y eficiencia al proceso: automatiza tareas, minimiza errores manuales y genera informes estructurados de forma ágil. Asimismo, opera en un espacio de mercado poco atendido (entre hojas de cálculo muy básicas y plataformas empresariales complejas). Como resaltan los expertos de KPMG, las tecnologías de la información serán determinantes para producir reportes más fiables y eficientes a largo plazo. Por tanto, una herramienta digital diseñada para PYMEs no solo facilita el **cumplimiento normativo**, sino que se convierte en una aliada estratégica que potencia la transparencia y competitividad de las empresas en la nueva era ESG. [Carrillo, A. J. (2024, January 16)].

Capítulo 3. TECNOLOGÍAS SELECCIONADAS Y JUSTIFICACIÓN TÉCNICA

A continuación, se detalla la elección de las tecnologías empleadas para el desarrollo de la aplicación web de reporte no financiero conforme a estándares ESG definidos por las normas ESRS. Se explica la arquitectura **front-end** basada en HTML, CSS y JavaScript puro, sin el uso de librerías o frameworks externos, y se justifica técnicamente esta decisión en términos de simplicidad, independencia tecnológica, facilidad de uso, ligereza, mantenimiento y compatibilidad con navegadores modernos. Asimismo, se destaca que la aplicación es completamente funcional en entorno local sin necesidad de infraestructura backend, alineando la solución con criterios de eficiencia, accesibilidad y aplicabilidad práctica para pymes.

3.1 TECNOLOGÍAS EMPLEADAS

La aplicación se construye utilizando **HTML5** para la estructura y el contenido, **CSS** para la presentación visual, y **JavaScript** puro (también conocido como vanilla JS) para dotar de interactividad y controlar la lógica en el lado del cliente. Esta combinación de tecnologías web estándares proporciona la base de funcionamiento de la herramienta, aprovechando características modernas del HTML5 (como etiquetas semánticas, elementos multimedia y APIs del navegador) junto con hojas de estilo CSS para un diseño atractivo y responsive, y scripts JavaScript para manejar eventos de usuario, validaciones y actualización dinámica de contenidos sin recargar la página.

Es importante resaltar que **no se han empleado bibliotecas ni frameworks externos** en el desarrollo, tales como Bootstrap para CSS o jQuery/React/Angular para JS. Toda la funcionalidad se implementa con código propio en JavaScript estándar, complementado por HTML5 y CSS3. Esta decisión de **no depender de frameworks** externos se tomó

deliberadamente para garantizar simplicidad e independencia tecnológica. Al trabajar con HTML, CSS y JavaScript puros, se evita la complejidad adicional y la curva de aprendizaje asociada a los frameworks, a la vez que se elimina la necesidad de cargar librerías voluminosas que podrían ralentizar la carga de la página. De este modo, la aplicación esencialmente consiste en archivos estáticos que cualquier navegador moderno puede interpretar directamente, lo que resulta en tiempos de carga más rápidos y una experiencia de usuario más fluida.

A continuación, se resumen las razones técnicas que respaldan la selección de estas tecnologías básicas y la exclusión de marcos externos:

- **Simplicidad de desarrollo y uso:** La utilización de tecnologías web fundamentales (HTML, CSS, JS) mantiene el proyecto sencillo de comprender y desarrollar. Incluso ingenieros o desarrolladores con conocimientos básicos web pueden contribuir sin tener que dominar la sintaxis o paradigmas de un framework. Esto reduce el periodo de aprendizaje y facilita el comienzo del desarrollo. Además, **los frameworks pueden introducir complejidad innecesaria:** a menudo implican una configuración y un aprendizaje profundos, mientras que con una aproximación vanilla es posible lograr la funcionalidad deseada con mayor claridad y control. Para proyectos de alcance acotado (como es este reporte para pymes), construir desde cero es una opción viable que otorga control total al desarrollador sobre cada componente. La sencillez en la base tecnológica también se traduce en una **interfaz más intuitiva para el usuario final**, al evitar sobrecargas o comportamientos opacos; cada interacción del usuario inicia código JavaScript explícitamente diseñado para esa acción, facilitando la trazabilidad y el ajuste fino de la experiencia. [Dom. (2020, August 24)].
- **Independencia tecnológica y mínima dependencia de terceros:** Al prescindir de librerías externas, la aplicación **evita dependencias** que podrían generar conflictos de versiones, problemas de compatibilidad o necesidad de actualizaciones forzadas. El código resultante es autónomo y no está atado a ciclos de vida de productos de

terceros, lo que asegura una mayor **longevidad y control** sobre la solución. En pequeños proyectos, las dependencias externas suelen ser innecesarias y hasta contraproducentes, ya que pueden añadir overhead sin aportar beneficios proporcionales. Esta independencia aporta también **seguridad y estabilidad**: el equipo de desarrollo conoce completamente el código que se ejecuta (al no incluir componentes de los que se desconozca su implementación interna) y no depende de la comunidad o empresa mantenedora de un framework para corregir fallos. Asimismo, al basarse en estándares abiertos, no existe riesgo de **bloqueo tecnológico** (vendor lock-in); cualquier desarrollador con conocimientos de estándares web puede mantener o extender la aplicación, sin importar herramientas propietarias.

- **Ligereza y eficiencia en rendimiento:** Una aplicación construida con HTML, CSS y JS puros es generalmente más **ligera** en términos de carga de recursos que una equivalente apoyada en un framework. Al no tener que cargar archivos de biblioteca pesados ni funcionalidades no utilizadas, se reduce el tamaño total de la página y la cantidad de scripts a ejecutar. Esto redundará en **mejores tiempos de carga y respuesta**, incluso en equipos modestos o conexiones lentas. En efecto, un sitio web estático con **menos “piezas móviles”** es mucho más fácil de optimizar en rendimiento, tendiendo a ser ágil y de carga rápida si está bien construido. Se elimina el overhead que impone un framework, logrando que la aplicación resultante sea muy **eficiente en el uso de recursos** del navegador. Desde la perspectiva de la ingeniería industrial, esta eficiencia es crucial: equivale a optimizar un proceso para que consuma menos energía y recursos, traducido aquí en menor consumo de CPU, memoria y datos, lo cual mejora la experiencia del usuario y la escalabilidad. Adicionalmente, el **rendimiento percibido** se beneficia de la interacción inmediata que permite JavaScript en cliente (por ejemplo, mostrando/ocultando secciones al instante) sin esperas de comunicación con servidor. Una página estática bien diseñada puede manejar incrementos moderados de tráfico sin problemas gracias a esta ligereza inherente, lo que favorece la

experiencia de múltiples usuarios concurrentes en un entorno pyme. [Kinsta. (2025, February 28)].

- **Facilidad de mantenimiento y evolución:** Mantener un código **simple y plano** suele ser más sencillo a largo plazo. En una aplicación vanilla, el desarrollador trata directamente con HTML para la estructura, CSS para la presentación y JS para la lógica, sin abstracciones adicionales. Esto hace que la **depuración de errores sea más directa**, pues todo ocurre en una sola capa y es más fácil seguir el flujo de la aplicación. Igualmente, para **actualizar o extender** la funcionalidad, no es necesario revisar documentación de terceros o adaptar el código a nuevas versiones de un framework; basta con modificar o añadir código nativo según se requiera. Esta característica es especialmente valiosa en pymes, donde el personal técnico puede ser limitado: el código autodocumentado y claro reduce la dependencia de desarrolladores externos. En sitios web sencillos que no requerirán adiciones funcionales frecuentes ni grandes cambios, una base vanilla ofrece un **mínimo overhead de mantenimiento** y evita preocupaciones por cambios drásticos en librerías externas. Mientras el estándar web (HTML/JS/CSS) permanezca vigente, y cabe destacar que son estándares muy maduros y estables, el sitio seguirá funcionando de forma consistente.
- **Compatibilidad universal con navegadores modernos:** Al ceñirse a estándares web (HTML5, CSS3 y ECMAScript), la aplicación es compatible de forma inmediata con **todos los navegadores modernos y plataformas** comunes (Chrome, Firefox, Safari, Edge, Opera). Los navegadores actuales han implementado plenamente HTML5, lo que significa que pueden interpretar correctamente las etiquetas y APIs que utiliza la aplicación. No se depende de funcionalidades propietarias ni de polyfills complejos; por tanto, cualquier usuario con un navegador actualizado podrá acceder al reporte sin inconvenientes adicionales. Esto también asegura la **adaptabilidad multi-dispositivo**: gracias a las características de HTML5 y CSS3, la interfaz se puede visualizar correctamente en distintos tamaños de pantalla (desde ordenadores de escritorio hasta tablets o

móviles) manteniendo la funcionalidad. La decisión de no usar *frameworks* front-end contribuye en este aspecto, ya que evita atarse a versiones específicas de navegadores o requerir características experimentales; en su lugar, se emplean elementos soportados de forma consistente en el amplio espectro de entornos usados en la actualidad. Desde un punto de vista industrial, esta compatibilidad equivale a diseñar un producto según normas estandarizadas, garantizando su funcionamiento en distintos contextos sin necesidad de ajustes particulares para cada caso. [Wolf. (2025, January 26)].

- **Accesibilidad mejorada mediante semántica web:** Construir la interfaz con HTML5 permite aprovechar la **semántica** propia del lenguaje (encabezados <header>, secciones <section>, artículos <article>, tablas <table>, formularios <form>) de modo correcto. Un principio fundamental en desarrollo web es que gran parte del contenido se puede hacer accesible simplemente **utilizando los elementos HTML con el propósito correcto** en todo momento. En este proyecto, se siguen las buenas prácticas de accesibilidad desde la base: por ejemplo, empleando etiquetas apropiadas para listas, encabezados jerárquicos para secciones, texto alternativo en imágenes, entre otros. Al no usar componentes personalizados de un framework (que a veces requieren esfuerzos adicionales para igualar la accesibilidad de los elementos nativos), nos aseguramos de que **los usuarios con discapacidades** o que utilicen tecnologías asistivas (lectores de pantalla, navegación solo con teclado) puedan interpretar y recorrer la página sin obstáculos. La ligereza de la página también mejora la accesibilidad en entornos con **limitaciones técnicas** (equipos antiguos o conexiones lentas), dado que reduce los tiempos de espera y las posibilidades de fallos. En esencia, la adopción de estándares abiertos no solo facilita el cumplimiento de criterios de accesibilidad (como WCAG), sino que también está alineada con el enfoque ESG, ya que promueve la inclusión y usabilidad universal de la herramienta.
- **Funcionamiento autónomo sin infraestructura backend:** Una de las ventajas más notables de la solución propuesta es que **la aplicación funciona**

completamente en entorno local o estático, sin requerir un servidor de aplicaciones ni base de datos para generar o servir el contenido. Esto significa que el reporte ESG puede **abrirse directamente en un navegador web como un archivo HTML local** o alojarse en cualquier servicio estático básico. Dado que no existe procesamiento del lado del servidor, la carga sobre infraestructura es mínima: el servidor (cuando se usa uno para publicar el sitio) se limita a entregar archivos estáticos, sin ejecutar lógica de negocio ni consultas a base de datos, lo cual **reduce drásticamente la complejidad y costo de despliegue**. Incluso es posible prescindir totalmente de un servidor web tradicional, utilizando por ejemplo un servicio de CDN o GitHub Pages para alojar los archivos, o simplemente compartiendo el conjunto de archivos HTML/CSS/JS a los interesados. Para una pyme, esto representa una **solución técnica muy viable**: no se necesita invertir en infraestructura ni en mantenimiento de servidores, y el riesgo de fallos por esa parte es nulo. Otra consecuencia beneficiosa es la **seguridad**: al no haber componentes de servidor ni bases de datos, la superficie de ataque se reduce significativamente; un sitio estático es inherentemente menos vulnerable a ataques comunes como inyección SQL y explotación de sesiones, puesto que **no hay lógica servidor que explotar**. Si bien siempre se debe asegurar el código cliente contra posibles vulnerabilidades de XSS u otras, la eliminación del backend simplifica enormemente el panorama de seguridad. Adicionalmente, la aplicación podría **funcionar sin conexión a internet** una vez descargada, dado que todos los recursos (contenido y scripts) residen en el cliente. Los componentes en el lado del cliente desarrollados con HTML, CSS y JS **no requieren un sistema operativo específico ni conectividad constante para operar**; en otras palabras, representan por sí mismos la aplicación web y pueden correr en cualquier navegador estándar independientemente de la plataforma. Esta capacidad *offline* significa que una pyme podría, por ejemplo, abrir su reporte ESG en una computadora aislada durante una presentación o enviarlo en un medio digital a un auditor, con la confianza de que todas las funciones (navegación interna, cálculos en JS) estarán disponibles sin necesidad de entorno adicional. De manera sucinta, la ausencia de *backend* no solo

simplifica la entrega, sino que aporta **robustez y portabilidad** a la solución. [Jose. (2022, September 16)].

En conjunto, las razones expuestas reflejan una **justificación técnica sólida** para la elección de HTML5, CSS y JavaScript puro en el desarrollo de esta aplicación de reporte ESG. Desde la óptica técnica, la solución destaca por su **eficiencia** (uso óptimo de recursos y rendimiento elevado), **confiabilidad** (menos dependencias = menos puntos de fallo), **mantenimiento simplificado** (estructura clara y modular), **accesibilidad** (alineada con estándares universales) y **adaptabilidad**. Todo ello se traduce en una herramienta práctica, de bajo costo y **apta para pymes**, que logra el objetivo propuesto, el cual es facilitar el reporte no financiero conforme a ESRS, sin incurrir en complejidades innecesarias ni replicar soluciones comerciales sobredimensionadas. Se proporciona así una solución tecnológica ligera pero robusta, fácil de implementar y operar en entornos empresariales con recursos limitados, manteniendo al mismo tiempo la calidad y el rigor que un reporte ESG requiere.

Capítulo 4. ESTRUCTURA MODULAR Y

ARQUITECTURA DE LA APLICACIÓN

La aplicación se organiza de forma modular en múltiples archivos HTML, cada uno con una responsabilidad definida, manteniendo coherencia en diseño y navegación. En la versión actual se presenta un **prototipo funcional parcial**: funcionalmente sólo están operativas las secciones de **Temas Clave** (Estándares ESG) y **Medición de indicadores**. En cambio, las secciones “Resultados” y “Generar Informe” aparecen en la barra de navegación como parte del diseño previsto para la versión avanzada del software, pero en esta versión inicial figura como enlaces inactivos sin funcionalidad implementada. En otras palabras, dichos módulos existen solamente a nivel de interfaz en la demo actual; su lógica interna y contenido efectivo se desarrollarán en la versión completa de pago.

La arquitectura modular descrita permite en el futuro agregar fácilmente nuevos módulos (por ejemplo, para gobernanza, medioambiente detallado, generación de informes) sin alterar la funcionalidad existente.

4.1 ORGANIZACIÓN MODULAR EN ARCHIVOS HTML

Se ha optado por consolidar gran parte de la interfaz en un único archivo central (`index.html`) siguiendo un enfoque de aplicación de página única o **SPA (Single Page Application)**.

Este archivo `index.html` actúa no solo como punto de entrada del sistema, sino también como contenedor principal de las distintas vistas internas de la aplicación. La estructura interna del documento incluye múltiples secciones identificadas mediante elementos `<section>` con IDs específicos: `introduccion`, `temas`, `medicion`, `resultados` e `informe`. Cada una de estas secciones representa una etapa clave del proceso de reporte no financiero

basado en el estándar ESRS, y su visibilidad se gestiona dinámicamente mediante JavaScript, sin necesidad de recargar la página.

La navegación entre estas secciones se realiza de forma fluida a través de funciones que muestran u ocultan las vistas relevantes, empleando el patrón SPA para mejorar la experiencia de usuario y reducir los tiempos de carga. Asimismo, se ha implementado el uso del objeto `sessionStorage` del navegador para almacenar temporalmente la sección activa. Esta funcionalidad permite que, al regresar desde páginas externas (por ejemplo, un informe PDF o enlaces de ayuda), el usuario sea redirigido automáticamente a la sección en la que se encontraba anteriormente, preservando así la continuidad de uso. Este comportamiento se analiza con mayor detalle en la sección 5.5.

A continuación, se introducen los principales archivos que conforman la aplicación y su función:

- **index.html (inicio):** Es el archivo principal y representa la **página de inicio o estructura matriz** de la aplicación. En index.html se define la estructura base de la interfaz, incluyendo elementos comunes como la cabecera con el título del reporte, menús de navegación generales y áreas de contenido donde se mostrará la información de los demás módulos. Actúa como **portal central** desde el cual el usuario puede acceder a las distintas secciones del reporte ESG. En términos de arquitectura, este archivo es el punto de entrada de la aplicación y mantiene el esqueleto común (layout) que se replica o utiliza en las demás vistas.
- **esrs2.html (temas clave):** Corresponde a un **módulo de contenido específico**, en este caso relacionado con las normas ESRS. Este archivo contiene la estructura y contenido particular de esa sección del reporte no financiero. Incluye texto descriptivo, tablas y indicadores ESG definidos por el estándar ESRS 2, con su propio marcado HTML semántico. Aunque se introduce aquí el nombre de archivo para claridad, nos referiremos a él simplemente como **módulo de estándares ESG** en lo sucesivo. Su función es proveer al usuario la información detallada de esa parte del estándar, en una página separada pero coherente en diseño con el resto.

- **esrs2m.html (medición):** Es otro **módulo HTML dedicado a las mediciones** ESG incluidas en el reporte, en este caso de esrs2. Contiene formularios para ingresar datos cuantitativos y mostrar indicadores clave de rendimiento (KPIs) ambientales, sociales o de gobernanza. Este archivo encapsula la lógica de presentación de la sección de medición, separándola de otras secciones más narrativas. Lo llamaremos de aquí en adelante **módulo de mediciones**. Al igual que el anterior, sigue la misma línea de estilo y estructura general, pero enfocado a funcionalidades (como cálculos client-side mediante JavaScript y gráficos embebidos si los hubiera) propias de la temática de métricas.
- **Otros archivos HTML (módulos adicionales):** La arquitectura está preparada para incorporar más archivos HTML para otras secciones relevantes del reporte ESG. Por ejemplo, módulos de resultados y de generar informe, para acabar de generar el producto demandado por las nuevas Políticas de sostenibilidad, Gobernanza corporativa e Información medioambiental detallada cada uno en su propio HTML (gobernanza.html, medioambiente.html, entre otros). Aunque estos no se mencionan explícitamente, el diseño modular permite agregarlos fácilmente siguiendo el mismo patrón: cada uno contendría el contenido y estructura de su sección, manteniendo consistencia con la interfaz general. En la planificación de la aplicación se consideró la **posibilidad de expansión**, por lo que la adición de nuevos módulos no altera la funcionalidad de los existentes, solo extiende el alcance de la aplicación.

Figura 4: Organización modular de la página web



Fuente: Elaboración propia

Todos estos archivos comparten una **coherencia en diseño y navegación**. Normalmente, para evitar duplicación de código, se define una plantilla común (el mismo fragmento de menú y pie de página) que cada HTML incluye ya sea manualmente o mediante pequeñas inserciones de código JS. De esta forma, el usuario percibe una experiencia unificada, aunque técnicamente esté navegando entre distintos documentos HTML. Cabe destacar que el uso de múltiples archivos estáticos **no implica pérdida de dinamismo**; por el contrario, cada módulo puede ser cargado o mostrado bajo demanda, tal como se describe en la siguiente sección.

4.2 NAVEGACIÓN CONTROLADA POR JAVASCRIPT ENTRE MÓDULOS

La interacción entre los módulos se implementa enteramente en el **lado del cliente** utilizando JavaScript, lo que proporciona una navegación fluida y controlada. En lugar de delegar la carga de nuevas páginas al navegador mediante enlaces tradicionales (lo cual produciría un refresco completo de la pantalla), la aplicación utiliza scripts para **interconectar los módulos HTML de forma dinámica**.

Concretamente, el archivo **index.html** incluye el **menú de navegación principal** que lista las secciones disponibles del reporte (opciones como "Inicio", "Estándares ESG", "Medición de indicadores"). Cada elemento de este menú está asociado a un manejador en JavaScript que responde al evento de clic del usuario. Cuando el usuario selecciona una sección, el código JS captura esa acción y ejecuta la lógica necesaria para mostrar el contenido correspondiente. Por ejemplo, al hacer clic en la sección de estándares ESG, en lugar de cargar una página nueva desde cero, el script podría cargar el contenido de **esrs2.html** dentro de una sección definida en **index.html** (mediante una petición fetch o insertando directamente elementos HTML predefinidos). De esta manera, la aplicación se comporta como una **Single Page Application (SPA) sencilla**, donde una barra de navegación controla una sección de vista que cambia según los ítems de menú seleccionados. Esta técnica permite **cambiar de sección sin recargar toda la página**,

haciendo la transición más rápida y suave visualmente para el usuario, ya que solo se actualiza la parte central de contenido.

Internamente, existen distintas formas de lograr esta navegación modular controlada por JavaScript. Una implementación posible es que index.html contenga un contenedor (por ejemplo, `<div id="contenido">`) donde se irá **inyectando** el contenido de los otros archivos HTML según la interacción del usuario. El JavaScript podría usar AJAX (Asynchronous JavaScript and XML) o la moderna API fetch para obtener el código HTML de esrs2.html, medicion.html..., e insertar ese fragmento en el contenedor, actualizando también el historial del navegador si se desea (por ejemplo, cambiando la URL con history.pushState para permitir marcadores o navegación con botones Atrás/Adelante). Alternativamente, cada módulo HTML podría estar incluido de antemano en la estructura, dentro de secciones ocultas que se muestran o esconden con JavaScript al navegar. Independientemente del método específico, el resultado es que la **lógica de navegación reside en el código cliente**, dando control total sobre qué mostrar en cada momento y manteniendo elementos comunes (como el menú o encabezado) visibles persistentemente.

El uso de JavaScript para la navegación aporta también la posibilidad de implementar características de usabilidad adicionales. Por ejemplo, se puede resaltar en el menú cuál es la sección activa actualmente, modificar dinámicamente el título de la página (para reflejar la sección en curso) o animar transiciones entre secciones, todo ello sin abandonar la página principal. Además, la **navegación horizontal por temas clave** mencionada se refiere a que el usuario no solo puede ir del índice general a una subsección, sino que desde una subsección puede saltar a otra de nivel similar fácilmente. Esto se logra manteniendo accesible el menú general (u otra barra de pestañas) en todo momento. En la práctica, significa que, si el usuario está en la sección de Medición de indicadores y desea consultar la sección de Estándares ESG, puede hacerlo con un solo clic en el menú, y el JavaScript cargará el contenido de esrs2.html en la vista, reemplazando el de medicion.html. **No es necesario volver primero a la página de inicio**, ya que la arquitectura proporciona rutas directas entre módulos hermanos mediante la interfaz común. Esta navegación transversal incrementa la **fluidez de uso**, permitiendo comparar o consultar diferentes apartados del

reporte rápidamente, característica valiosa en documentos de referencia como lo es un informe ESG.

Asimismo, cada módulo incluye en su contenido elementos para **retornar al índice principal** (un enlace o botón "Volver al inicio" situado al comienzo o final de la sección). Estos elementos, al ser activados, desencadenan también funciones JavaScript que restauran la vista inicial (mostrando nuevamente la panorámica general en index.html). Esto garantiza que el usuario nunca quede "atrapado" en una sección específica; siempre hay una ruta claramente señalizada de regreso a la visión global del reporte. En implementaciones sencillas, este retorno podría ser simplemente un enlace a index.html; sin embargo, con la navegación controlada por JS, se puede hacer de forma más elegante, preservando el estado de la página principal si fuera necesario. Por ejemplo, si el índice tenía alguna interacción (una lista de capítulos desplegada, etc.), el retorno vía JS podría recordar ese estado.

Es importante señalar que, aunque la navegación esté centrada en JavaScript, **la aplicación sigue siendo funcional incluso si el JavaScript estuviera deshabilitado**, al menos en un nivel básico. Esto es parte de una buena práctica de ingeniería web (mejora progresiva): los enlaces de menú podrían apuntar a las páginas HTML correspondientes de forma tradicional, de modo que, si por alguna razón el script no carga, el usuario aún podría navegar recargando la página completa a esrs2.html, esrs2m.html, etc. No obstante, la experiencia óptima y prevista es con JavaScript habilitado, aprovechando la dinámica descrita.

4.3 LÓGICA ESTRUCTURAL DE LA INTERFAZ Y FLUJO DE NAVEGACIÓN

Resumiendo, la estructura lógica: **index.html funciona como interfaz principal**, presentando la visión general del reporte y sirviendo de hub para la navegación. Desde esta interfaz, el usuario tiene **enlaces o botones** hacia los distintos submódulos (secciones ESG específicas y mediciones entre otros). Al activar cualquiera de esos enlaces, la **lógica**

JavaScript intercepta la acción y carga la sección solicitada dentro de la misma interfaz, actualizando el área de contenido. Una vez dentro de un **submódulo**, el usuario dispone tanto de opciones de navegación horizontal (el mismo menú principal para saltar a otra sección paralela) como de la opción de retorno al índice. El flujo típico sería:

1. Usuario abre la aplicación en **index.html** (ya sea como página web servida o archivo local).

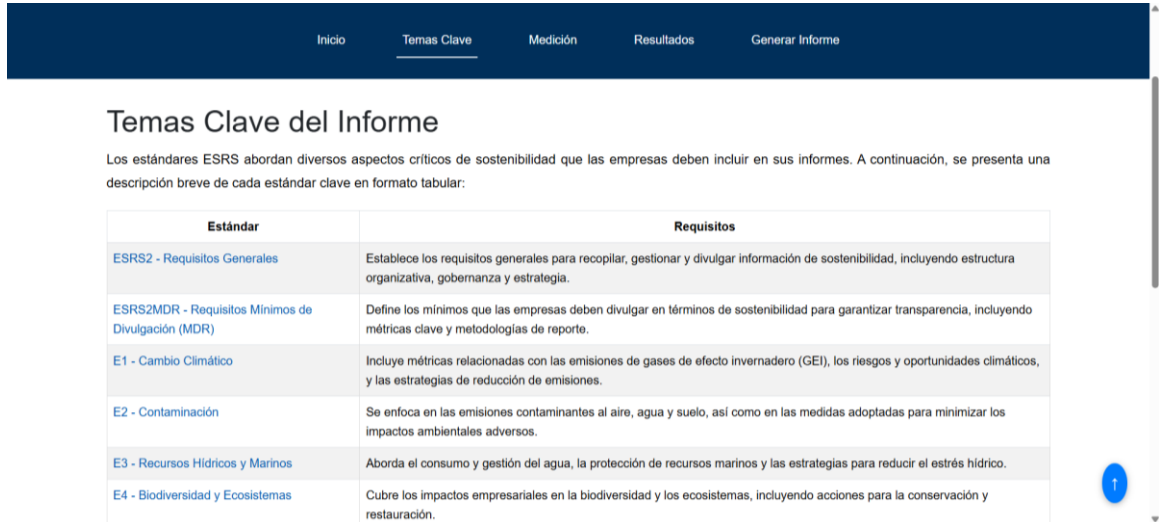
Figura 5: Etapa 1 de flujo típico



Fuente: Elaboración propia

2. Observa la página de inicio con introducción y menú de secciones disponibles (Temas clave, Medición, Resultados y Generar informe).
3. Hace clic en una sección, "**Temas clave**" tras haber visto la de inicio.

Figura 6: Etapa 2 de flujo típico



Temas Clave del Informe

Los estándares ESRS abordan diversos aspectos críticos de sostenibilidad que las empresas deben incluir en sus informes. A continuación, se presenta una descripción breve de cada estándar clave en formato tabular:

Estándar	Requisitos
ESRS2 - Requisitos Generales	Establece los requisitos generales para recopilar, gestionar y divulgar información de sostenibilidad, incluyendo estructura organizativa, gobernanza y estrategia.
ESRS2MDR - Requisitos Mínimos de Divulgación (MDR)	Define los mínimos que las empresas deben divulgar en términos de sostenibilidad para garantizar transparencia, incluyendo métricas clave y metodologías de reporte.
E1 - Cambio Climático	Incluye métricas relacionadas con las emisiones de gases de efecto invernadero (GEI), los riesgos y oportunidades climáticos, y las estrategias de reducción de emisiones.
E2 - Contaminación	Se enfoca en las emisiones contaminantes al aire, agua y suelo, así como en las medidas adoptadas para minimizar los impactos ambientales adversos.
E3 - Recursos Hídricos y Marinos	Aborda el consumo y gestión del agua, la protección de recursos marinos y las estrategias para reducir el estrés hídrico.
E4 - Biodiversidad y Ecosistemas	Cubre los impactos empresariales en la biodiversidad y los ecosistemas, incluyendo acciones para la conservación y restauración.

Fuente: Elaboración propia

- JavaScript carga el contenido de Temas clave en la vista actual (o navega a esa página de forma controlada). El menú destaca ahora "Temas clave" como sección activa.
- Dentro de la sección Temas clave, el usuario lee la información. Desde allí, decide pasar a "**Medición de indicadores**" mediante el menú o un botón "Siguiete sección", tras haber pulsado cualquiera de los temas clave y haber revisado el contenido necesario, opta por continuar con las mediciones.

Figura 7: Etapa 3 de flujo típico

Detalles del Estándar ESRS 2			
A continuación, se detallan los puntos de divulgación para el estándar ESRS 2 :			
Número	ESRS	Nombre	Tipo de Datos
1	BP-1	Divulgación de la base general para la preparación de la declaración de sostenibilidad	narrativa
2	BP-1	Base para la preparación de la declaración de sostenibilidad	semi-narrativa
3	BP-1	El alcance de la consolidación de la declaración de sostenibilidad consolidada es el mismo que el de los estados financieros	narrativa
4	BP-1	Indicación de las empresas subsidiarias incluidas en la consolidación que están exentas de la presentación de informes de sostenibilidad individuales o consolidados	narrativa
5	BP-1	Divulgación del alcance en que la declaración de sostenibilidad cubre la cadena de valor ascendente y descendente	narrativa
6	BP-1	Se ha utilizado la opción de omitir un dato específico correspondiente a la propiedad intelectual, know-how o resultados de innovación	semi-narrativa
7	BP-1	Se ha utilizado la opción permitida por el Estado miembro para omitir la divulgación de desarrollos inminentes o asuntos en curso de negociación	semi-narrativa
8	BP-2	Divulgaciones en relación con circunstancias específicas	narrativa
9	BP-2	Se ha desviado de los horizontes de tiempo medio o largo definidos por ESRS 1	semi-narrativa
10	BP-2	Divulgación de definiciones de horizontes de tiempo medio o largo	narrativa
11	BP-2	Divulgación de razones para aplicar diferentes definiciones de horizontes de tiempo	narrativa

Inicio	Temas Clave	Medición	Resultados	Generar Informe
--------	-------------	----------	------------	-----------------

Corporate Sustainability Reporting Directive

Medición

Los estándares ESRS abordan diversos aspectos críticos de sostenibilidad que las empresas deben incluir en sus informes. A continuación, se presenta una descripción breve de cada estándar clave en formato tabular:

Estándar	Requisitos
ESRS2 - Requisitos Generales	Establece los requisitos generales para recopilar, gestionar y divulgar información de sostenibilidad, incluyendo estructura organizativa, gobernanza y estrategia.
ESRS2MDR - Requisitos Mínimos de Divulgación (MDR)	Define los mínimos que las empresas deben divulgar en términos de sostenibilidad para garantizar transparencia, incluyendo métricas clave y metodologías de reporte.
E1 - Cambio Climático	Incluye métricas relacionadas con las emisiones de gases de efecto invernadero (GEI), los riesgos y oportunidades climáticos,

Fuente: Elaboración propia

- El sistema, otra vez vía JS, presenta el contenido de **Medicion.html** en el mismo marco, reemplazando el anterior, sin una recarga completa. El menú pasa a resaltar "Medición". En este marco, tiene la misma posibilidad de seleccionar de la tabla los estándares que desea rellenar y rellenarlo en el formato apropiado, ya indicado.

Figura 8: Etapa 4 de flujo típico

Detalles del Estándar ESRS 2				
A continuación, se detallan los puntos de divulgación para el estándar ESRS 2 :				
Número	ESRS	Nombre	Tipo de Datos	Respuesta
1	BP-1	Divulgación de la base general para la preparación de la declaración de sostenibilidad	narrativa	
2	BP-1	Base para la preparación de la declaración de sostenibilidad	semi-narrativa	
3	BP-1	El alcance de la consolidación de la declaración de sostenibilidad consolidada es el mismo que el de los estados financieros	narrativa	
4	BP-1	Indicación de las empresas subsidiarias incluidas en la consolidación	narrativa	

Fuente: Elaboración propia

- Finalmente, el usuario puede pulsar "**Volver al inicio**". JavaScript limpiará el contenido actual y mostrará de nuevo la pantalla inicial definida en *index.html* (o simplemente redirigirá a *index.html* si se optó por carga tradicional en esa interacción).

Figura 9: Etapa 5 de flujo típico

Inicio
Temas Clave
Medición
Resultados
Generar Informe

¿Qué es y por qué es importante la **CSRD**?

La **CSRD (Corporate Sustainability Reporting Directive)** es una reciente regulación de la Unión Europea creada con el objetivo de que las compañías muestren mayor transparencia y responsabilidad en relación a su efecto en el **ESG**, aspirando a que las compañías divulguen datos no financieros de forma más nítida, comprensible y exacta.

Esta normativa expande la aplicación de la reglamentación previa, la **NFRD**, incorporando a un mayor número de compañías que ahora tendrán que informar sobre su sostenibilidad. Además, define criterios más precisos para valorar cómo las acciones de las compañías afectan el ambiente y la sociedad, y cómo los factores ESG influyen en su rendimiento y los riesgos a los que se enfrentan. La CSRD implica un paso hacia una mayor claridad y responsabilidad en el mundo empresarial.

La aplicación de la CSRD es clave para establecer estándares uniformes de sostenibilidad en toda la **Unión Europea**, lo que facilita la comparación entre empresas y fomenta modelos de negocio más sostenibles, alineados con las expectativas de mercados globales responsables.

Asimismo, la CSRD incluye los Estándares Europeos de Información sobre Sostenibilidad (**ESRS**), que proporcionan un marco detallado para la elaboración de informes de sostenibilidad. Esto asegura que las empresas divulguen información precisa, relevante y comparable sobre sus prácticas ESG.

Fuente: [Consilium](#)

Fuente: Elaboración propia

Este **diseño modular e interactivo** ofrece una experiencia cohesiva: pese a estar técnicamente dividido en archivos, el sitio **se comporta como una única aplicación integrada**. Para el usuario, las transiciones son rápidas y la estructura consistente; para el desarrollador, cada sección está aislada en su módulo, lo que facilita la organización interna del código.

4.4 ESCALABILIDAD, MANTENIMIENTO Y ADAPTACIÓN FUTURA DE LA ARQUITECTURA

La elección de una arquitectura modular basada en múltiples archivos HTML interconectados no solo satisface las necesidades actuales del proyecto, sino que también prepara el terreno para su **evolución futura**. En términos de ingeniería de software, esta aproximación sigue el principio de *alta cohesión y bajo acoplamiento*: cada módulo (archivo HTML + sus scripts asociados) concentra contenido funcionalmente relacionado (cohesión), mientras que las interdependencias entre módulos están bien definidas a través de interfaces claras (navegación controlada) y se reducen al mínimo necesario (bajo acoplamiento). Esto redundará en beneficios significativos:

- **Facilidad para escalar funcionalidades:** Si en el futuro se desea ampliar el alcance del reporte incorporando nuevos apartados (como una nueva norma ESRS sectorial, o un capítulo de Benchmarking de indicadores), se puede simplemente **crear un nuevo módulo HTML** para esa sección y enlazarlo en el menú de navegación. Gracias al diseño modular, la adición de este nuevo componente no requerirá reestructurar la aplicación entera, sino solo **integrarlo** en puntos específicos (añadir un botón en el menú, quizás incluir algún script adicional si tiene lógica propia). Esto se alinea con las mejores prácticas de construir aplicaciones escalables: uno de los fundamentos para lograr escalabilidad es precisamente el diseño modular, dividiendo la aplicación en componentes independientes para facilitar su crecimiento y mantenimiento. En efecto, tener la aplicación dividida en **componentes o páginas independientes facilita la**

escalabilidad y el mantenimiento conforme la aplicación se desarrolla y crece.

Cada nuevo requerimiento puede abordarse en un módulo separado, lo que permite trabajar en paralelo sin interferir con otras partes, y probar cada sección de forma aislada. [Bridge. (2024, May 27)].

- **Mantenimiento localizado y actualización sencilla:** En una arquitectura modular, las actualizaciones o correcciones se circunscriben al módulo relacionado con la funcionalidad en cuestión. Por ejemplo, si se necesita modificar la forma en que se calculan ciertos indicadores ESG, es probable que esos cálculos residan en `medicion.html` o en su script asociado; el desarrollador puede ajustar ese módulo sin temor a romper funcionalidades de la sección de estándares, puesto que no hay una mezcla de ambas lógicas en el mismo archivo. Esto hace que la aplicación sea más **fácil de mantener y depurar**, ya que el código está organizado por temas. Además, la consistencia en la estructura de cada archivo (gracias a la plantilla común de `index.html`) significa que aprender a mantener un módulo permite mantenerlos todos con un esfuerzo incremental mínimo. Desde la perspectiva de la calidad, es más sencillo **probar y verificar** partes individuales de la aplicación (como verificar que `esrs2.html` cumple con los requisitos de contenido ESRS) antes de integrarlas, lo que se traduce en un desarrollo más controlado y menos propenso a errores globales. Esta característica modular recuerda a la segmentación en líneas de producción: cada estación (módulo) realiza su tarea y se puede ajustar sin detener toda la fábrica. Del mismo modo, aquí cada archivo se puede mejorar o arreglar sin tener que reescribir la aplicación completa, lo cual es un enfoque robusto frente a cambios.
- **Reutilización y consistencia:** La arquitectura propuesta permite también **reutilizar componentes entre módulos**, reforzando la consistencia. Por ejemplo, si se desarrolla un fragmento de código JavaScript para validar la entrada de datos de indicadores, ese mismo código puede incluirse en varios módulos que lo necesiten, manteniendo la misma lógica en toda la aplicación. Dado que no dependemos de un framework que dicte la estructura, hemos podido definir

convenciones internas: estructuras HTML similares, clases CSS comunes y funciones JS reutilizables. Esto no solo reduce duplicación de código, sino que facilita que los comportamientos sean uniformes. Para una pyme, esta consistencia significa que el **mantenimiento futuro** (quizá realizado por diferentes personas) se realiza sobre un patrón reconocible. Asimismo, en caso de actualizar el estilo visual corporativo, contar con una hoja CSS central vinculada a todos los módulos permite cambiar la apariencia en un punto único y propagarlo en todas las secciones.

- **Adaptación a futuras versiones de ESRS:** Dado que el proyecto se enmarca en un entorno normativo (estándares ESG/ESRS) que puede evolucionar con el tiempo, era crucial diseñar la aplicación de manera adaptable. La **arquitectura modular facilita la incorporación de cambios regulatorios**. Por ejemplo, si en versiones futuras de las normas ESRS se agregan nuevos requisitos de divulgación, se puede integrar un nuevo módulo que contenga esos requerimientos adicionales, o ajustar los existentes sin rehacer el sistema completo. La separación por temas hace que, si cambia la norma **en la parte ambiental**, solo el módulo ambiental deba actualizarse; si se introducen nuevos KPIs, se agrega o modifica el módulo de mediciones, y así sucesivamente. Incluso si la estructura de los reportes no financieros se reconfigura (agrupando secciones de otra forma), la aplicación puede reordenar o renombrar módulos con relativa facilidad, manteniendo la columna vertebral (index + navegación) estable. En términos técnicos, hemos diseñado con la **extensibilidad** en mente: la aplicación podrá crecer en contenidos y funcionalidades alineados con ESRS sin necesidad de una refactorización radical. Esto protege la inversión de desarrollo a futuro, un aspecto importante en proyectos industriales donde la **vida útil** de las soluciones y su capacidad de actualizarse conforme a nuevas exigencias normativas es un factor de éxito.
- **Paralelismo con arquitecturas industriales modulares:** Desde una óptica de ingeniería industrial, es válido hacer un paralelismo: así como en sistemas físicos complejos se prefiere una arquitectura modular (con las máquinas o procesos divididos en unidades funcionales, que pueden mejorarse o reemplazarse

individualmente), aquí se aplica el mismo principio al software. La modularidad añade **robustez** al sistema: un problema en una sección (módulo) no colapsa toda la aplicación, y las mejoras pueden implementarse incrementalmente. Por ejemplo, si se detecta un error únicamente en la página de medición, los demás módulos siguen operativos e independientes; corregir ese error es acotado en alcance. Esta aislación de fallos potenciales mejora la **confiabilidad global** del aplicativo, un atributo muy valorado en proyectos de ingeniería.

En síntesis, la estructura modular y la arquitectura implementada proporcionan una solución técnica ordenada y **escalable**. Se ha dividido la funcionalidad en unidades lógicas (index.html y módulos de contenido especializados) que interactúan mediante una navegación controlada por JavaScript, logrando un balance entre simplicidad y potencia. Esta arquitectura cumple con los objetivos iniciales de ofrecer una herramienta viable para pymes, ya que es fácil de mantener y adaptar sin necesidad de grandes recursos, y a la vez ofrece la **rigurosidad necesaria** para abarcar los estándares ESRS presentes y futuros. La claridad en la organización del código y la robustez de la navegación aseguran que la aplicación pueda crecer en alcance (más secciones, más interacción) sin perder estabilidad ni facilidad de uso. En suma, la solución presentada no solo resuelve el problema inmediato (crear una página web para reporte ESG), sino que lo hace con una visión a largo plazo, garantizando que el **diseño técnico** esté preparado para incorporar las **exigencias venideras** en materia de sostenibilidad empresarial

Capítulo 5. NAVEGACIÓN E INTERACTIVIDAD CON JAVASCRIPT

5.1 INTRODUCCIÓN

A continuación, se describe la implementación de la lógica de navegación e interactividad de la plataforma de reporte no financiero basada en los estándares **ESRS**. El sistema se ha desarrollado con HTML, CSS y JavaScript, adoptando una estructura de página única o single-page application (SPA) simplificada. Esto significa que la navegación entre las distintas secciones del informe no implica recargar completamente la página, sino simplemente mostrar u ocultar secciones del contenido dinámicamente. Para lograr esta funcionalidad, se ha implementado una función clave llamada `showSection(id)`, que alterna la visibilidad de las secciones mediante la manipulación de clases CSS (por ejemplo, asignando la clase `active` a la sección seleccionada). Adicionalmente, se han incorporado características de interactividad como el desplazamiento automático (scroll) suave hacia la sección activa utilizando `scrollIntoView`, así como el uso de `sessionStorage` para preservar el estado de navegación entre sesiones de usuario o al realizar saltos a otras páginas relacionadas.

A continuación, se detalla el funcionamiento de la navegación interna, incluyendo la estructura HTML/CSS empleada, la lógica de la función `showSection(id)`, la gestión de eventos del DOM (como la carga de la página y los clics de usuario) y el mecanismo de persistencia de la navegación mediante almacenamiento de sesión. Se ilustrará esta explicación con fragmentos de código comentados y capturas de pantalla de la interfaz en funcionamiento, presentadas con el rigor y claridad esperados en un contexto de ingeniería.

5.2 ESTRUCTURA DE NAVEGACIÓN TIPO SPA

La aplicación web se comporta como una SPA (single-page application), en la que todas las secciones de contenido residen en un único documento HTML y se muestran u ocultan según la interacción del usuario, sin recargar la página entera. Para ello, el archivo HTML principal contiene un elemento `<main>` que agrupa múltiples secciones `<section>`, cada una correspondiente a una vista o apartado distinto del informe (Introducción, Temas Clave, Medición, Resultados, Generar Informe). Por ejemplo, el código HTML simplificado a continuación muestra la definición de dos secciones del informe: la sección de introducción y la sección "Temas Clave". La primera se marca inicialmente con la clase CSS `active` para que sea visible por defecto al cargarse la página, mientras que la segunda carece de dicha clase (permaneciendo oculta hasta que el usuario navegue hacia ella):

Figura 10: Ejemplo de estructura HTML con secciones diferenciadas y clase active para navegación entre bloques

```
html
<section id="introduccion" class="active">
  <h1>¿Qué es y por qué es importante la CSRD?</h1>
  <p>La <strong>CSRD</strong> es una reciente regulación de la Unión Europea
creada con el objetivo de que las compañías muestren mayor transparencia y
responsabilidad en materia de sostenibilidad...</p>
  <!-- ... contenido adicional ... -->
</section>

<section id="temas">
  <h2>Temas Clave del Informe</h2>
  <p>Los estándares ESRS abordan diversos aspectos críticos de
sostenibilidad que las empresas deben incluir en sus informes. A
continuación, se presenta una descripción breve de cada estándar clave
en formato tabular:</p>
  <!-- ... contenido de la sección Temas, e.g., tabla de estándares
ESRS ... -->
</section>
```

Fuente: Elaboración propia

En la estructura anterior, la sección con `id="introduccion"` está marcada con `class="active"`, lo que indica que es la sección actualmente activa (visible). Las demás

secciones (como la de `id="temas"`) no tienen la clase activa y, por definición en la hoja de estilos, permanecen ocultas. En la hoja CSS del proyecto se define el comportamiento de estas clases; por ejemplo, se puede usar una regla para que las secciones inactivas no se muestren (`display: none`) y solo la sección con clase `active` se visualice (`display: block` o equivalente). De este modo, en cualquier instante solo existe una sección principal visible, proporcionando la ilusión de cambiar de página sin efectuar recargas completas.

La navegación entre secciones se realiza a través de un menú superior fijo en la página (dentro de la cabecera `<header>`). Este menú contiene enlaces o botones para cada sección principal del informe. En la implementación, cada elemento de navegación está definido con un vínculo que invoca a la función JavaScript `showSection(id)` al hacer clic, pasando el identificador de la sección de destino. Por ejemplo, el menú puede estar codificado de la siguiente forma:

Figura 11: Fragmento de código con enlaces a secciones mediante onclick y showSection

```
html
<nav>
  <ul>
    <li><a href="#" onclick="showSection('introduccion')">Inicio</a></li>
    <li><a href="#" onclick="showSection('temas')">Temas Clave</a></li>
    <li><a href="#" onclick="showSection('medicion')">Medición</a></li>
    <li><a href="#" onclick="showSection('resultados')">Resultados</a></li>
    <li><a href="#" onclick="showSection('informe')">Generar
Informe</a></li>
  </ul>
</nav>
```

Fuente: Elaboración propia

En este fragmento, cada elemento `<a>` del menú tiene un atributo `onclick` que llama a `showSection(...)` con el id de la sección correspondiente. El `href="#"` se utiliza para que el enlace sea navegable sin especificar una URL nueva (opcionalmente podría emplearse `event.preventDefault()` dentro de la función para evitar el comportamiento por defecto del enlace, aunque en este caso el enlace `#` no recarga la página). Gracias a esta configuración, cuando el usuario pulsa una opción del menú (por ejemplo "Temas Clave"),

se ejecuta la función `showSection('temas')` que hará visible la sección de Temas Clave y ocultará las demás, actualizando el contenido central de la página de forma instantánea.

Figura 12: Interfaz de la plataforma mostrando la sección "Temas Clave del Informe"



Fuente: Elaboración propia

En el menú de navegación superior (barra azul), la pestaña **Temas Clave** se resalta indicando la sección actual (subrayado en este caso), mientras que el contenido correspondiente (incluyendo el título de sección y la tabla de estándares ESRS) se muestra en la zona principal.

Como se aprecia en la Figura 6, al activar la sección "Temas Clave", el menú superior destaca dicha sección (mediante un estilo visual diferente, en este caso subrayado o cambio de color) para indicar al usuario en qué parte del informe se encuentra. Simultáneamente, en el área de contenido principal aparece el título Temas Clave del Informe, seguido de una explicación y una tabla con los estándares ESRS y sus descripciones, mientras que el contenido de las demás secciones (Introducción, Medición...) permanece oculto. La experiencia para el usuario es fluida: la interfaz responde al clic mostrando la nueva sección de inmediato, sin flashes ni recargas completas, lo que mejora la interacción y la sensación de continuidad.

Es importante señalar que esta funcionalidad se apoya únicamente en HTML, CSS y JavaScript puro, sin utilizar frameworks SPA externos. Esto permite un mayor control sobre la navegación y un rendimiento adecuado dado el alcance de la aplicación. En el siguiente apartado, se detalla la implementación de la función JavaScript que hace posible este comportamiento de mostrar/ocultar secciones de forma dinámica.

5.3 IMPLEMENTACIÓN DE LA FUNCIÓN *SHOWSECTION(ID)*

La función `showSection(id)` constituye el núcleo de la navegación interactiva de la plataforma. Su responsabilidad es sencilla pero fundamental: dada la cadena `id` de una sección, la función oculta todas las secciones existentes y luego muestra únicamente aquella cuyo atributo `id` coincide con el parámetro recibido. De este modo, garantiza que solo una sección esté visible a la vez. A continuación, se presenta el código de dicha función, con comentarios que explican cada paso de la lógica implementada:

Figura 13: Código de la función `showSection` para la navegación entre secciones

```
javascript
function showSection(id) {
    // Obtener todas las secciones dentro del elemento <main>
    const sections = document.querySelectorAll('main section');
    // Remover la clase 'active' de todas las secciones para ocultarlas
    sections.forEach(section => section.classList.remove('active'));

    // Obtener la sección objetivo por su identificador
    const targetSection = document.getElementById(id);
    // Si la sección objetivo existe, agregar la clase 'active' para mostrarla
    if (targetSection) {
        targetSection.classList.add('active');
    }
}
```

Fuente: Elaboración propia

Como se observa, `showSection` primero selecciona **todas** las secciones del contenido principal (`document.querySelectorAll('main section')`) y a cada una le remueve la clase `active` (línea en la que se itera con `forEach`). Esto asegura que ninguna sección previa permanezca visible. Luego, obtiene la sección destino mediante

`document.getElementById(id)`; si dicha sección se encuentra en el DOM, se le añade la clase CSS `active`. En consecuencia, la sección destino pasa a estar visible mientras que las demás quedan ocultas, cumpliendo con el comportamiento deseado de navegación exclusiva por secciones.

Es relevante mencionar que la clase `active` es interpretada por la hoja de estilos CSS para controlar la visibilidad. Una posible implementación en el CSS (simplificada) sería:

Figura 14: Reglas CSS para mostrar y ocultar secciones mediante la clase `active`

```
css
/* Ejemplo de reglas CSS para mostrar/ocultar secciones */
section {
    display: none;           /* Ocultar todas las secciones por defecto */
}
section.active {
    display: block;         /* Mostrar solo la sección que tenga la clase
'active' */
}
```

Fuente: Elaboración propia

De esta forma, la llamada a `section.classList.add('active')` en JavaScript cambia el estilo de esa sección en particular para hacerla visible en la página.

Además de manejar la visibilidad del contenido, `showSection` contribuye a la actualización de la interfaz de usuario, por ejemplo, al resaltar en el menú la sección activa. En la implementación, esto se logra también mediante clases CSS: al cambiar de sección, se podría actualizar la clase de los elementos del menú correspondientes (añadiendo, por ejemplo, una clase `active` al botón del menú seleccionado y removiéndola de los demás) para que el estilo activo en el menú siga al contenido mostrado. Aunque en el código proporcionado arriba solo se maneja la clase de las secciones, un enfoque coherente se ha aplicado al menú. En la práctica, el marcado HTML del menú incluye estilos específicos para el elemento activo, aprovechando por ejemplo la pseudo-clase: `focus` o la manipulación directa de clases mediante JavaScript adicional, de forma que, al hacer clic en una opción, ésta quede visualmente marcada.

5.4 DESPLAZAMIENTO SUAVE Y MANEJO DE EVENTOS DEL DOM

Para mejorar la experiencia de usuario, se implementó también una funcionalidad de **desplazamiento automático suave** hacia la sección seleccionada. Esto resulta útil en casos donde el cambio de sección implica moverse a una parte distinta de la página (por ejemplo, si el contenido tiene cierta longitud o si el usuario ha hecho **scroll** hacia abajo). Al activar una nueva sección, especialmente si ésta está más arriba en la estructura del DOM, conviene asegurarse de que el inicio de la sección aparezca en pantalla sin exigir que el usuario haga scroll manualmente.

En JavaScript, el método `Element.scrollIntoView()` permite desplazar la vista hasta un elemento dado. En este proyecto, se utiliza con la opción `{behavior: "smooth"}` para lograr una transición suave en lugar de un salto brusco. A modo ilustrativo, se empleó el siguiente código para detectar ciertas condiciones al cargar la página y para realizar el scroll:

Figura 15: Código para desplazamiento suave y detección de hash en el evento

```
javascript
document.addEventListener("DOMContentLoaded", function () {
    // Si la URL contiene el hash #temas al cargar, hacer scroll suave a esa
    sección
    if (window.location.hash === "#temas") {
        document.getElementById("temas").scrollIntoView({ behavior: "smooth" });
    }
});
```

Fuente: Elaboración propia

El fragmento anterior demuestra el uso de un event listener sobre el evento `DOMContentLoaded` del documento. En cuanto el contenido HTML se ha cargado y parseado por completo, la función anónima proporcionada comprueba si la URL actual contiene un hash (fragmento identificador) correspondiente a la sección "temas". Esto podría ocurrir si, por ejemplo, el usuario navegó directamente a `index.html#temas` o si volvió a la página principal con un marcador específico. De cumplirse la condición, la función obtiene el elemento `<section id="temas">` y llama a `scrollIntoView({behavior: "smooth"})`,

provocando que la página se desplace animadamente hasta el inicio de dicha sección. Así, el usuario es llevado directamente al contenido relevante de forma elegante.

De manera análoga, se podría invocar `scrollIntoView` cuando se llama a `showSection(id)` tras un evento de clic en el menú o botón "Siguiente". Por ejemplo, para garantizar que la nueva sección activa comienza en la parte superior de la ventana gráfica, podría implementarse dentro de `showSection` o inmediatamente después de invocarla, un código como:

Figura 16: Ejemplo de invocación directa para mejorar la navegación entre secciones

```
javascript
targetSection.scrollIntoView({ behavior: "smooth" });
```

Fuente: Elaboración propia

Esto complementaría el cambio de visibilidad con un desplazamiento automático, eliminando saltos abruptos en la navegación. En la práctica, dado que la aplicación suele mostrar las secciones en un contenedor principal fijo, el cambio de contenido normalmente coloca el nuevo encabezado de sección en una posición similar, por lo que el desplazamiento puede no ser crítico al hacer clic en el menú superior (que ya está en la cabecera). Sin embargo, en interacciones de botones internos (como el botón "Siguiente" al final de una sección, que lleva a la siguiente sección), el scroll suave sí añade valor: al pulsarlo, la siguiente sección aparece y la vista se mueve gradualmente hacia arriba, manteniendo el contexto.

La gestión de eventos en este proyecto se realiza tanto mediante controladores en línea (atributos `onclick` en elementos HTML, como se vio en el menú) como mediante la asignación de escuchadores con `addEventListener` en el código JavaScript. El ejemplo anterior mostró un uso de `addEventListener` para el evento `DOMContentLoaded`. Asimismo, se emplea `window.onload` en otro contexto para ejecutar código cuando la página termina de cargar completamente (incluyendo recursos como imágenes). En todos estos casos, la

idea es encapsular la lógica que debe ocurrir en respuesta a un suceso del ciclo de vida de la página o de la interacción del usuario.

Un caso específico manejado con eventos del DOM es la funcionalidad de **botón "Siguiente"**, **"Como medir cada métrica"** y **"Resultados"**. Al final de ciertas secciones, se ha incorporado un botón que permite al usuario avanzar secuencialmente a la siguiente sección del informe sin tener que subir al menú. Este botón está vinculado igualmente a la función `showSection` para la sección siguiente. Por ejemplo, al final de la sección de Introducción (id `"introduccion"`), un botón podría definirse así:

Figura 17: Botón "Siguiente" con navegación secuencial entre secciones mediante

```
html
<button onclick="showSection('temas')">Siguiente</button>
```

Fuente: Elaboración propia

Este control desencadena la navegación a la sección "Temas Clave" directamente. Opcionalmente, se podría complementar con una llamada a `scrollIntoView` para asegurarse de que al cargar "Temas Clave" el usuario comienza viendo el inicio de esa sección. En la implementación actual, la experiencia ya es continua puesto que la sección previa se oculta y la nueva ocupa su lugar en el flujo del documento; no obstante, añadir el desplazamiento suave contribuye a la percepción de una transición natural.

En términos concisos, mediante la combinación de **manipulación del DOM** (mostrando/ocultando secciones con clases CSS via JavaScript) y **control de eventos** (capturando clics y eventos de carga para ejecutar lógica adicional como el scroll), la aplicación ofrece una navegación interactiva e intuitiva. En la siguiente sección se discute cómo se conserva el estado de esa navegación incluso cuando el usuario abandona temporalmente la página o refresca la sesión.

5.5 PERSISTENCIA DEL ESTADO DE NAVEGACIÓN CON *SESSIONSTORAGE*

Un requerimiento importante en aplicaciones web interactivas es mantener la continuidad cuando el usuario realiza acciones como recargar la página o navegar hacia contenidos relacionados y luego regresar. En el contexto de esta plataforma de reporte, se decidió utilizar **Session Storage** (API de almacenamiento web de sesión) para guardar información mínima sobre la última sección visitada o la necesidad de volver a una sección específica. A diferencia de Local Storage, que persiste indefinidamente hasta ser limpiado, el SessionStorage almacena datos que perduran solo durante la sesión de navegador actual (es decir, mientras el navegador o pestaña permanezca abierta). Esto encaja bien con la idea de retener el estado entre saltos cortos sin generar datos permanentes.

En la práctica, cuando el usuario navega a una sección externa o realiza una acción que lo lleva fuera de la página principal (por ejemplo, ver el detalle de un estándar ESRS en una página aparte), el sistema puede almacenar una marca en `sessionStorage` indicando qué sección debe mostrarse de nuevo al regresar. Por ejemplo, si desde la sección "Temas Clave" el usuario accede a una vista detallada de uno de los estándares (cargando otra página HTML con la información ampliada), la aplicación establece una clave de sesión del tipo `"volver_a_temas"` antes de salir. De este modo, cuando el usuario eventualmente regresa a la página principal, un script al cargar leerá esa clave y activará automáticamente la sección correspondiente.

El siguiente fragmento de código demuestra cómo se implementa esta lógica al cargar la página principal:

Figura 18: Persistencia de la sección activa usando sessionStorage al recargar o regresar

```
javascript
document.addEventListener("DOMContentLoaded", function () {
    if (sessionStorage.getItem("volver_a_temas") === "true") {
        sessionStorage.removeItem("volver_a_temas"); // Borrar el estado para
        no reutilizarlo
        showSection('temas'); // Mostrar directamente la sección Temas Clave
    }
});
```

Fuente: Elaboración propia

En este código, al completarse la carga del DOM, se verifica si existe en el `sessionStorage` una clave `"volver_a_temas"` con valor `"true"`. De ser así, significa que la última interacción del usuario antes de recargar o volver a esta página indicó que se deseaba regresar a la sección "Temas Clave". El script entonces borra esa marca (para evitar efectos indeseados si se recarga nuevamente más tarde) y llama a `showSection('temas')`, forzando la visualización de la sección Temas Clave inmediatamente. Gracias a esto, el usuario ve directamente la sección de interés, en lugar de, por ejemplo, la página de inicio o introducción por defecto, preservando así el contexto de navegación.

Se aplicó un mecanismo similar para otros posibles saltos. Por ejemplo, si desde la sección de "Medición" el usuario navegaba a otra funcionalidad (digamos, la generación de un informe en PDF en una nueva pestaña), la aplicación podía establecer `sessionStorage.setItem("volver_a_medicion", "true")`. Al retornar a la interfaz web, un código comprobaba esta condición:

Figura 19: Código para retener la navegación entre secciones tras volver a la página

```
javascript
window.onload = function() {
    if (sessionStorage.getItem("volver_a_medicion") === "true") {
        console.log("Regresaste desde medición"); // (Mensaje de depuración)
        sessionStorage.removeItem("volver_a_medicion");
        showSection('medicion'); // (En caso de implementarse, mostraría la
        sección Medición)
    }
};
```

Fuente: Elaboración propia

El código anterior incluye un mensaje de registro en consola a modo de depuración, y conceptualmente podría invocar `showSection('medicion')` para reactivar la sección de Medición al volver.

Con este uso de `SessionStorage`, el sistema logra un **mínimo de persistencia**: mantiene solamente flags muy concretos sobre navegación, suficientes para restaurar el estado de la interfaz en escenarios acotados. Esto mejora la usabilidad, ya que evita que el usuario tenga que recordar y volver a navegar manualmente a la última sección en que estaba trabajando. Al mismo tiempo, al limitar la persistencia a la sesión actual, se asegura que al iniciar una nueva sesión (por ejemplo, al día siguiente o desde otro dispositivo) la aplicación comience en su estado inicial predeterminado, evitando confusión con estados antiguos.

En la implementación definitiva, se ha verificado que el sistema utiliza correctamente `sessionStorage` para preservar el estado de navegación. Al cargar la página principal (`index.html`), se comprueba si existe una clave previamente almacenada con el identificador de la sección activa. En caso afirmativo, se invoca automáticamente la función `showSection('temas')` (o equivalente) para restaurar la vista correspondiente.

Además, se actualiza el hash de la URL cuando se cambia de sección, y se aplica `scrollIntoView()` para garantizar que el encabezado de la nueva sección quede visible desde el primer momento. Aunque este detalle no modifica sustancialmente la funcionalidad principal, mejora la precisión visual de la navegación, especialmente en dispositivos móviles o pantallas pequeñas.

Esta lógica asegura una experiencia continua al volver desde módulos externos (por ejemplo, desde un PDF del informe generado) sin perder el punto del flujo en el que se encontraba el usuario.

Es importante destacar que esta solución de persistencia es sencilla y apropiada para la escala del proyecto. Alternativas más complejas podrían incluir parámetros en la URL (como el fragmento hash ya mencionado) o incluso almacenamiento en servidor del

progreso del usuario. Sin embargo, para una demo `SessionStorage` ofrece un equilibrio adecuado entre simplicidad de implementación y mejora de la experiencia de navegación.

5.6 MEJORA DE LA EXPERIENCIA VISUAL CON ANIMACIONES

Para mejorar la experiencia de usuario y reforzar la percepción de fluidez de la plataforma, se han incorporado dos librerías externas de animación: AOS (Animate On Scroll) y GSAP (GreenSock Animation Platform). Estas herramientas permiten aplicar transiciones visuales suaves que enriquecen la navegación y facilitan la interpretación del contenido.

En concreto, AOS se utiliza para animar los elementos del DOM al hacer scroll, empleando atributos como `data-aos="fade-up"` o `data-aos="zoom-in"` para definir efectos de entrada. Por otro lado, GSAP se encarga de animar títulos y cabeceras de sección mediante scripts de control que permiten secuencias más complejas (por ejemplo, deslizamiento o desvanecimiento con temporización personalizada).

Estas mejoras no alteran la estructura funcional del sistema, pero contribuyen significativamente a una **percepción más profesional e intuitiva** del flujo del reporte.

5.7 CONCLUSIONES DE LA IMPLEMENTACIÓN DE NAVEGACIÓN

Se ha presentado detalladamente cómo la plataforma implementa la navegación e interactividad mediante JavaScript, logrando una experiencia fluida propia de una SPA ligera. La función `showSection(id)` actúa como pilar central, permitiendo alternar el contenido visible de forma dinámica y controlada a través de clases CSS, sin recargar la página. La estructura basada en secciones HTML y un menú de navegación superior facilita al usuario el acceso a las distintas partes del informe de manera intuitiva.

Se han incorporado mejoras en la experiencia de uso, como el desplazamiento automático suave al enfocar una nueva sección, contribuyendo a una transición agradable entre contenidos. Asimismo, la gestión de eventos del DOM (ejemplificada con el uso de

`addEventListener` para detectar la carga del documento y las interacciones de los usuarios) ha permitido integrar lógica adicional como la comprobación de *flags* de `SessionStorage` o la activación de desplazamientos sin ensuciar el marcado HTML y manteniendo un código organizado.

Por último, la utilización de `SessionStorage` para la persistencia temporal del estado de navegación asegura que el usuario pueda retomar su trabajo donde lo dejó tras interacciones que lo llevan fuera de la página principal, todo ello sin almacenar información sensible ni permanente, sino solo indicadores contextuales durante la sesión activa.

En conjunto, la navegación e interactividad implementadas cumplen con los objetivos de proveer una interfaz **ágil, moderna y user-friendly** para la plataforma de reportes no financieros. Estas características técnicas se han desarrollado con un enfoque profesional y alineado con los estándares del proyecto, asegurando que la solución propuesta no solo funcione correctamente, sino que también ofrezca una experiencia positiva al usuario final. La elección de una solución basada en JavaScript puro demuestra la capacidad de diseñar e implementar sistemas eficientes y a medida, adecuados a las necesidades del proyecto, manteniendo la claridad del código y la facilidad de mantenimiento, aspectos fundamentales en un desarrollo de calidad a nivel de ingeniería industrial.

Capítulo 6. ENTRADA DE DATOS Y DISEÑO DE FORMULARIOS

A continuación se detalla el diseño de la interfaz de entrada de datos y formularios de la aplicación, centrándose en aspectos técnicos clave: el uso de tablas HTML como estructura principal para los formularios, la habilitación de campos editables mediante el atributo `contenteditable`, la lógica implementada para gestionar los datos introducidos (manipulación del **DOM**, almacenamiento temporal y eventos), y las consideraciones de usabilidad en la interfaz (claridad visual, botones de navegación, organización lógica de la información). Asimismo, se describe la integración de una barra de progreso que informa al usuario del porcentaje completado dentro de cada formulario. Finalmente, se presentan propuestas de mejora, como la validación de entradas, el almacenamiento persistente de los datos (por ejemplo, mediante `localStorage` o una base de datos) y la exportación del informe generado a formato PDF, entre otras.

6.1 FORMULARIO EN LA SECCIÓN DE INTRODUCCIÓN

Aunque este capítulo se centra en los formularios de medición ESG, la página inicial **incluye un pequeño formulario introductorio** (los tres inputs numéricos y el botón “¿Estoy obligado?”) en la página inicial. Se añadió este formulario preliminar de evaluación de umbrales, indicando que todavía no forma parte de los cálculos principales pero que se usará para verificar obligaciones de reporte. Se puede observar en la figura 3.

6.2 USO DE TABLAS COMO INTERFAZ PRINCIPAL PARA LA ENTRADA DE DATOS

Para la entrada de datos se ha optado por utilizar **tablas HTML** como interfaz principal de los formularios. Cada estándar ESRS relevante del informe no financiero se presenta al usuario como una tabla donde cada **punto de divulgación** (requisito específico que debe reportarse) ocupa una fila. En dichas filas se estructuran varias columnas informativas: un identificador numérico o código del requisito, la referencia al estándar (por ejemplo, **E1** para Clima), una descripción breve del contenido a divulgar, el tipo de dato esperado (narrativo, porcentaje, cuantitativo...), y finalmente un campo donde el usuario ingresa su **respuesta**. Esta disposición en formato tabular permite **organizar la información de manera clara y estructurada**, facilitando la lectura y el ingreso de datos, ya que cada pregunta queda visualmente alineada con su espacio de respuesta correspondiente. Además, las tablas proveen un entorno familiar para muchos usuarios (similar a hojas de cálculo), lo que mejora la usabilidad al reducir la curva de aprendizaje para introducir la información requerida.

Figura 20: Interfaz de entrada de datos basada en una tabla para el estándar ESRS E1

Detalles del Estándar ESRS E1				
Número	ESRS	Nombre	Tipo de Datos	Respuesta
1	E1	Divulgación de cómo se tienen en cuenta las consideraciones relacionadas con el clima en la remuneración de los miembros de los órganos administrativos, de gestión y supervisión	Narrativa	La política retributiva de Nova S.A. incluye indicadores clave de desempeño (KPIs) relacionados con objetivos climáticos y de sostenibilidad. Un 20%
2	E1	Porcentaje de la remuneración reconocida que está vinculada a consideraciones relacionadas con el clima	Porcentaje	20 %
3	E1	Explicación de las consideraciones relacionadas con el clima que se tienen en cuenta en la remuneración de los miembros de los órganos administrativos, de gestión y supervisión	Narrativa	Las consideraciones climáticas incluidas en la remuneración variable se basan en tres ejes: (1) reducción anual del 5% en emisiones de GEI de
4	E1	Divulgación del plan de transición para la mitigación del cambio climático	Narrativa	Nova ha desarrollado un plan de transición climática 2025-2035 con una hoja de ruta basada en ciencia (SBTi). Incluye inversiones por valor de 1.200
5	E1	Explicación de cómo los objetivos son compatibles con la limitación del calentamiento global a un grado y medio en línea con el Acuerdo de París	Narrativa	Los objetivos de Nova están alineados con el escenario de 1,5 °C del IPCC y se han validado con la metodología SBTi. Se presenta una reducción del 40%

Fuente: Elaboración propia

La tabla presenta en columnas el código y nombre del estándar, la descripción del requisito, el tipo de dato esperado y la **respuesta editable** proporcionada por el usuario. En la parte superior se observa la barra de progreso asociada a este formulario, que indica el grado de avance en el llenado. Como se aprecia, el uso de una tabla asegura que todas las preguntas y respuestas estén alineadas y visibles de un vistazo, lo que ayuda al usuario a comprender qué datos debe suministrar en cada caso sin perder contexto.

6.3 IMPLEMENTACIÓN DE CAMPOS EDITABLES MEDIANTE CONTENTEDITABLE

Con el fin de permitir que el usuario ingrese datos directamente en las celdas de la tabla, se implementaron **campos editables** utilizando el atributo HTML5 `contenteditable`. Este atributo global, cuando se establece en `true`, indica que el elemento debería ser editable por el usuario, haciendo que el navegador cambie su presentación para permitir la edición del contenido. En la práctica, se habilitó la propiedad `contenteditable="true"` sobre las celdas de la columna de **Respuesta** en cada tabla de formulario, transformando esas celdas en áreas editables similares a campos de formulario tradicionales (como inputs de texto o textareas). De este modo, el usuario puede hacer clic en la celda y escribir o modificar directamente su respuesta allí mismo, sin necesidad de ventanas emergentes ni formularios separados, logrando una experiencia de entrada de datos más **fluida e intuitiva**.

Cabe destacar que, a diferencia de los elementos de formulario convencionales, los elementos con `contenteditable` no envían sus datos mediante un mecanismo de envío automático, por lo que es necesario manejar manualmente la extracción y gestión de la información introducida. No obstante, esta aproximación ofrece una **gran flexibilidad** en el diseño de la interfaz, al permitir formatear y estructurar el contenido a voluntad (por ejemplo, mantener el texto dentro de la tabla) sin las limitaciones de algunos controles de formulario. La elección de `contenteditable` resultó adecuada dado que los formularios requieren principalmente texto libre narrativo o valores numéricos simples; además, facilita aplicar estilos CSS personalizados a las celdas editables para indicar su estado (por

ejemplo, resaltarlas cuando están vacías o en foco) y mejorar la experiencia de usuario. En resumen, el atributo `contenteditable` ha permitido crear campos de respuesta **in situ** dentro de las tablas, manteniendo la consistencia visual del formulario y habilitando la edición directa del contenido de forma natural para el usuario.

6.4 LÓGICA DE GESTIÓN DE LOS DATOS INTRODUCIDOS (DOM, ALMACENAMIENTO TEMPORAL, EVENTOS)

Desde el punto de vista funcional, la aplicación implementa una lógica en **JavaScript** del lado del cliente para gestionar los datos introducidos y la navegación a través de los formularios. Dado que se trata de una aplicación HTML autónoma (sin un backend inmediato), los datos proporcionados por el usuario permanecen en el **Document Object Model (DOM)** de la página mientras se completan los formularios. Es decir, el contenido de cada campo editable reside en la estructura HTML actual y puede ser accedido y manipulado mediante scripts. Para consolidar la información introducida o transferirla entre secciones, se emplean estructuras de almacenamiento temporal en el navegador. En particular, se ha utilizado la **Web Storage API** (por ejemplo, `sessionStorage`) para guardar ciertos estados de la aplicación de forma transitoria mientras el usuario navega entre páginas o secciones. Este almacenamiento en el navegador permite **preservar información breve durante la sesión**, como indicadores de qué sección estaba activa o ciertos datos intermedios, evitando pérdidas de información si el usuario se mueve entre distintos apartados de la aplicación.

Asimismo, la aplicación hace uso extensivo de **eventos del DOM** para responder a las acciones del usuario y cambios en la interfaz. Por ejemplo, es posible capturar el evento de entrada (`input`) en los campos editables para detectar cuando el usuario modifica una respuesta, desencadenando así funciones que actualizan inmediatamente la información interna o la barra de progreso. También se utilizan eventos de carga de la página, como `DOMContentLoaded`, para inicializar el estado de los formularios o restaurar datos si es necesario. La lógica implementada se encarga de actualizar dinámicamente la interfaz

conforme a las acciones: por ejemplo, cada vez que el usuario avanza al siguiente formulario, se podría guardar el progreso actual en `sessionStorage` y luego, al retornar, reconstruir o posicionar la vista en el punto adecuado.

Figura 21: Fragmento para mantener la sección activa tras volver a la página

```
javascript
<script>
    document.addEventListener("DOMContentLoaded", function () {
        if (sessionStorage.getItem("volver_a_temas") === "true") {
            sessionStorage.removeItem("volver_a_temas"); // Borrar el estado
            showSection('temas'); // Mostrar directamente la sección Temas
        }
    });
</script>
```

Fuente: Elaboración propia

En este ejemplo, al producirse el evento `DOMContentLoaded`, la función comprueba si existe un indicador almacenado (clave `volver_a_temas`) que señale que el usuario regresó a la sección "Temas" tras haber visitado otra sección. De ser así, se elimina dicha marca de la sesión y se invoca a `showSection('temas')` para mostrar automáticamente la sección correspondiente. Este mecanismo garantiza una **experiencia continua**: si el usuario vuelve desde otro módulo, la aplicación recuerda y presenta la última sección relevante, evitando que tenga que navegar manualmente de nuevo hasta ella. Una vez utilizada, la marca es eliminada del almacenamiento temporal para no repetir la acción en futuras cargas de página.

En cuanto a la **manipulación del DOM**, la función `showSection(id)` (encargada de mostrar una sección de formulario específica) es un buen ejemplo de la lógica implementada. Al ser llamada con el identificador de sección, esta función localiza todos los elementos `<section>` del formulario principal y les remueve una clase CSS activa (por ejemplo, `.active`), ocultándolos; luego añade dicha clase únicamente a la sección objetivo, haciéndola visible. De esta forma, mediante simples operaciones DOM (agregar/quitar clases), se controla qué formulario o apartado está visible en cada momento, manteniendo la interfaz limpia y enfocada. Combinado con el uso de eventos y almacenamiento temporal antes descritos, este enfoque permite gestionar eficientemente la navegación multi-sección

y el mantenimiento de los datos introducidos por el usuario mientras interactúa con la aplicación.

6.5 INTERFAZ ORIENTADA A LA USABILIDAD: CLARIDAD VISUAL, NAVEGACIÓN Y ORGANIZACIÓN

En el diseño de la interfaz se ha priorizado la **usabilidad** y la claridad visual, de modo que el usuario pueda comprender fácilmente la información presentada y navegar por los formularios sin dificultad. La información se organiza de forma **lógica y secuencial**, siguiendo la estructura natural del reporte no financiero. Por ejemplo, primero se presentan los temas clave o secciones generales (requisitos generales, medioambientales, sociales, etc.), para luego profundizar en cada estándar específico con sus respectivos formularios de entrada de datos. Cada sección o formulario está claramente rotulada con títulos descriptivos y, cuando corresponde, con breves instrucciones o contexto para guiar al usuario en la introducción de los datos. Asimismo, se ha mantenido un diseño limpio, utilizando tipografías legibles, contrastes adecuados y una disposición equilibrada de los elementos en la pantalla, evitando la saturación de información. La consistencia visual entre secciones (mismos estilos para tablas, botones y textos) ayuda a que el usuario **se familiarice rápidamente** con la interfaz y anticipe cómo interactuar en cada paso.

Figura 22: Pantalla de Temas Clave del Informe

Inicio
Temas Clave
Medición
Resultados
Generar Informe

Temas Clave del Informe

Los estándares ESRS abordan diversos aspectos críticos de sostenibilidad que las empresas deben incluir en sus informes. A continuación, se presenta una descripción breve de cada estándar clave en formato tabular:

Estándar	Requisitos
ESRS2 - Requisitos Generales	Establece los requisitos generales para recopilar, gestionar y divulgar información de sostenibilidad, incluyendo estructura organizativa, gobernanza y estrategia.
ESRS2MDR - Requisitos Mínimos de Divulgación (MDR)	Define los mínimos que las empresas deben divulgar en términos de sostenibilidad para garantizar transparencia, incluyendo métricas clave y metodologías de reporte.
E1 - Cambio Climático	Incluye métricas relacionadas con las emisiones de gases de efecto invernadero (GEI), los riesgos y oportunidades climáticos, y las estrategias de reducción de emisiones.
E2 - Contaminación	Se enfoca en las emisiones contaminantes al aire, agua y suelo, así como en las medidas adoptadas para minimizar los impactos ambientales adversos.
E3 - Recursos Hídricos y Marinos	Aborda el consumo y gestión del agua, la protección de recursos marinos y las estrategias para reducir el estrés hídrico.
E4 - Biodiversidad y Ecosistemas	Cubre los impactos empresariales en la biodiversidad y los ecosistemas, incluyendo acciones para la conservación y restauración.
E5 - Uso de Recursos y Economía Circular	Analiza el uso eficiente de los recursos, la gestión de residuos y la transición hacia modelos de economía circular.
S1 - Mano de Obra Propia	Se centra en las condiciones laborales, diversidad e inclusión, desarrollo profesional y bienestar de los empleados de la empresa.
S2 - Trabajadores en la Cadena de Valor	Examina el impacto de las actividades empresariales en los trabajadores de la cadena de suministro, incluyendo derechos laborales y condiciones de trabajo.
S3 - Comunidades Afectadas	Evalúa cómo las operaciones empresariales afectan a las comunidades locales y las medidas para abordar impactos negativos.
S4 - Consumidores y Usuarios Finales	Considera la seguridad y sostenibilidad de los productos y servicios, así como la protección de los derechos de los consumidores.
G1 - Conducta Empresarial	Incluye políticas de ética empresarial, anticorrupción, cumplimiento normativo y transparencia en la gestión corporativa.

Anterior
Cómo medir cada métrica

Fuente: Elaboración propia

Pantalla de **Temas Clave del Informe**, que ilustra la organización lógica de la información antes de ingresar a los formularios detallados.

En esta vista se muestra una tabla resumen con los distintos **estándares ESRS** aplicables (ESRS 2, ESRS 2 MDR, ESRS E1, E2...), junto con una breve descripción de cada bloque

temático. Esta lista permite al usuario identificar las áreas que deberá reportar y navegar hacia los formularios específicos de cada estándar.

Figura 23: Pantalla de que ilustra el menú superior de navegación.



Fuente: Elaboración propia

La interfaz presenta un menú superior de navegación (como se observa en la parte superior: Inicio, Temas Clave, Medición, Resultados, Generar Informe) y utiliza títulos y tablas para brindar una **visión general estructurada** del reporte. Al mantener la información organizada de esta manera, el usuario puede orientarse fácilmente en qué parte del proceso de reporte se encuentra y qué secciones vienen a continuación.

Figura 24: Botones de navegación del menú Temas clave.



Fuente: Elaboración propia

Un aspecto fundamental de la usabilidad incorporado en la aplicación es la presencia de **botones de navegación** claramente identificados (por ejemplo, botones "Anterior" o enlaces en el menú) que permiten avanzar o retroceder entre las secciones del formulario. Estos controles de navegación están ubicados de forma consistente (típicamente al final de cada sección o en la barra de menú superior) para que el usuario siempre sepa cómo continuar al siguiente paso o volver al anterior. La lógica detrás de estos botones, como se describió anteriormente, muestra u oculta secciones mediante funciones JavaScript dedicadas. A continuación, se presenta un pequeño fragmento de código que ejemplifica esta funcionalidad:

Figura 25: Función utilizada para la navegación entre formularios.

```
javascript
function showSection(id) {
    const sections = document.querySelectorAll('main section');
    const links = document.querySelectorAll('nav ul li a');

    sections.forEach(section => section.classList.remove('active'));
    links.forEach(link => link.classList.remove('active-link'));

    const targetSection = document.getElementById(id);
    if (targetSection) targetSection.classList.add('active');

    history.replaceState(null, '', `#${id}`); // ACTUALIZA LA URL //
    window.scrollTo(0, 0);

    links.forEach(link => {
        if (link.getAttribute('onclick')?.includes(id)) {
            link.classList.add('active-link');
        }
    });
}
```

Fuente: Elaboración propia

El código con la función `showSection(id)` selecciona todas las secciones del contenido principal (`<main>`) y les remueve la clase CSS activa, luego añade la clase `.active` solo a la sección cuyo identificador coincide con el parámetro proporcionado (`id`). De esta forma, al asociar esta función a los controles de navegación (como al evento de clic de un botón "Anterior" o a la selección de un elemento del menú), la interfaz **cambia de forma fluida** al siguiente formulario o sección pertinente, sin recargar la página completa. Este enfoque

mejora la usabilidad al brindar **transiciones suaves** y mantener el estado del formulario, permitiendo al usuario concentrarse en una sección a la vez sin distraerse con información de otras partes del formulario.

6.6 INTEGRACIÓN DE UNA BARRA DE PROGRESO EN LOS FORMULARIOS

Para mejorar la experiencia del usuario durante el llenado de formularios extensos, la aplicación incorpora una **barra de progreso** visual que indica el porcentaje de completitud dentro de cada formulario o sección. Esta barra de progreso ofrece al usuario una referencia inmediata sobre su avance, estableciendo expectativas claras y aportando una sensación de logro a medida que completa campos del formulario. Al poder **visualizar cuánto le falta para terminar**, el usuario se siente más tranquilo y motivado para continuar, evitando que el proceso se perciba como interminable. Diversos estudios de usabilidad sugieren que este tipo de indicadores pueden calmar a los usuarios y reducir la incertidumbre durante tareas largas, disminuyendo el riesgo de abandono. En el contexto de los formularios ESRS, la barra de progreso refuerza la transparencia del proceso: el usuario sabe en todo momento qué porcentaje del formulario ha completado y cuánto le resta por responder, lo cual mejora la **confianza y satisfacción** con la herramienta.

Figura 26: Definición de estilos CSS para la barra de progreso de los formularios

```
css
.progress-container {
    width: 100%;
    background-color: #e0e0e0;
    border-radius: 10px;
    margin: 20px 0;
}
.progress-bar {
    height: 10px;
    width: 0%;
    background-color: #28a745;
    border-radius: 10px;
    transition: width 0.3s;
}
```

Fuente: Elaboración propia

- `.progress-container`: define el espacio total de la barra.
- `.progress-bar`: representa el porcentaje completado, cuyo width se actualiza dinámicamente en función de los campos cumplimentados.

En la implementación desarrollada, la barra de progreso se sitúa en la parte superior de cada formulario, en un lugar visible pero que no distrae del contenido principal. Repetida a continuación:

Figura 27: Formulario con barra de progreso dinámica y visualización de cumplimiento

Detalles del Estándar ESRS E1				
A continuación, se detallan los puntos de divulgación para el estándar ESRS E1 :				
Número	ESRS	Nombre	Tipo de Datos	Respuesta
1	E1	Divulgación de cómo se tienen en cuenta las consideraciones relacionadas con el clima en la remuneración de los miembros de los órganos administrativos, de gestión y supervisión	Narrativa	La política retributiva de Nova S.A. incluye indicadores clave de desempeño (KPIs) relacionados con objetivos climáticos y de sostenibilidad. Un 20%
2	E1	Porcentaje de la remuneración reconocida que está vinculada a consideraciones relacionadas con el clima	Porcentaje	20 %
3	E1	Explicación de las consideraciones relacionadas con el clima que se tienen en cuenta en la remuneración de los miembros de los órganos administrativos, de gestión y supervisión	Narrativa	Las consideraciones climáticas incluidas en la remuneración variable se basan en tres ejes: (1) reducción anual del 5% en emisiones de GEI de
4	E1	Divulgación del plan de transición para la mitigación del cambio climático	Narrativa	Nova ha desarrollado un plan de transición climática 2025-2035 con una hoja de ruta basada en ciencia (SBTi). Incluye inversiones por valor de 1.200
5	E1	Explicación de cómo los objetivos son compatibles con la limitación del calentamiento global a un grado y medio en línea con el Acuerdo de París	Narrativa	Los objetivos de Nova están alineados con el escenario de 1,5 °C del IPCC y se han validado con la metodología SBTi. Se necesita una reducción del 40%

Fuente: Elaboración propia

A nivel técnico, se optó por emplear la etiqueta semántica `<progress>` provista por HTML5 para representar dicha barra. La etiqueta `<progress>` está diseñada para **visualizar el progreso de una tarea**, mostrando gráficamente una barra cuyo llenado depende de los valores de progreso actuales. En este caso, se configura el atributo `max` del `<progress>` al número total de campos o preguntas del formulario, y el atributo `value` al número de campos ya completados. Un script de JavaScript se encarga de actualizar estos valores dinámicamente conforme el usuario introduce sus respuestas. Por ejemplo, si un formulario contiene 10 elementos de datos y el usuario ha rellenado 7 de ellos, el código ajustará `value=7` (con `max=10`), lo que hará que la barra de progreso muestre un **70%** de avance

(incluso es posible mostrar ese porcentaje numérico dentro de la barra para mayor claridad). Gracias a esta actualización automática en tiempo real, la barra de progreso refleja fielmente el **estado de avance** del usuario en cada sección, sirviendo como un **feedback visual inmediato** que guía y estimula al usuario hasta la finalización del formulario.

6.7 PROPUESTAS DE MEJORA

Si bien la aplicación cumple con los objetivos básicos de permitir la entrada de datos para el reporte no financiero basado en ESRS, se han identificado varias **propuestas de mejora** y extensiones futuras que podrían enriquecer su funcionalidad y robustez:

- **Validación de entradas:** Implementar validaciones más estrictas en los campos de respuesta. Por ejemplo, asegurar que los campos marcados como obligatorios no queden vacíos, que los campos de tipo numérico (como porcentajes o importes) contengan efectivamente datos numéricos en el formato correcto, o que las respuestas narrativas no excedan cierta longitud. Estas validaciones podrían realizarse en tiempo real, proporcionando **retroalimentación inmediata** al usuario (marcando en rojo o mostrando mensajes de error junto al campo) y evitando inconsistencias o errores en los datos recopilados. Una validación robusta mejora la calidad de la información final y la confiabilidad del informe generado.
- **Almacenamiento persistente de datos:** Incorporar mecanismos de **persistencia** que permitan guardar el progreso del usuario más allá de la sesión actual. Actualmente, los datos introducidos residen en el DOM y en `sessionStorage` de forma temporal (mientras la pestaña del navegador permanece abierta). Como mejora, se podría utilizar `localStorage` para mantener los datos en el navegador incluso si el usuario cierra y vuelve a abrir la aplicación, o mejor aún, implementar un módulo de backend con una base de datos donde se guarde el formulario parcialmente completado. De esta forma, un usuario podría retomar el llenado del informe en otro momento o desde otro dispositivo, recuperando la información ya introducida. El almacenamiento persistente también abriría la puerta a

funcionalidades multiusuario, permitiendo que varias personas colaboren en el llenado del mismo informe en distintas sesiones.

- **Exportación del informe a PDF u otros formatos:** Desarrollar la funcionalidad de **generación de informes en PDF** a partir de los datos ingresados. Esto permitiría que, tras completar los formularios, el usuario pueda obtener un documento PDF formateado conforme a un estándar presentable (como el logotipo de la empresa, cabeceras de sección, tablas de resultados), listo para ser entregado a los organismos correspondientes o integrado en la memoria anual de la empresa. Existen bibliotecas JavaScript (jsPDF, PDFMake) que podrían emplearse para convertir el contenido HTML dinámico en un PDF. Alternativamente, podría implementarse una plantilla predefinida que sea llenada con los datos y exportada. La exportación no solo a PDF sino también a formatos como **Excel** o **JSON** podría ser útil para que la empresa reutilice los datos en otros sistemas o análisis.
- **Mejoras en la usabilidad y funcionalidades adicionales:** Continuar refinando la interfaz incorporando principios de **diseño responsivo** para garantizar que la aplicación se visualice correctamente en distintos tamaños de pantalla (por ejemplo, en tablets o portátiles pequeños), dado que en un entorno empresarial los usuarios podrían querer revisar o completar el informe desde diversos dispositivos. También se podría habilitar una **navegación no lineal**, permitiendo al usuario saltar directamente a secciones específicas mediante un índice o menú lateral, manteniendo la barra de progreso general del informe. Otra posible mejora es incluir **indicadores visuales en el menú** o en la lista de temas clave (como marcas de verificación) que muestren qué secciones ya están completadas total o parcialmente, facilitando un control global del avance. Adicionalmente, se podrían incorporar ayudas contextuales, como pequeños textos de ayuda o ejemplos junto a cada pregunta, e incluso funcionalidades de **autoguardado** periódico para asegurar que ningún dato se pierda inesperadamente. Estas propuestas de mejora contribuirían a hacer la herramienta más robusta, user-friendly y alineada con las necesidades prácticas de las empresas al preparar sus reportes de sostenibilidad.

Capítulo 7. VISUALIZACIÓN DE RESULTADOS Y GENERACIÓN DEL INFORME

A continuación, se describe funcionalidades previstas para la versión avanzada de pago de la aplicación. En la demo actual **no están implementadas** las secciones de Visualización de Resultados y Generación de Informe: aparecen en el menú, pero como enlaces inactivos sin funcionalidad. Por ello, este capítulo se mantiene en la memoria únicamente como hoja de ruta futura para el desarrollo completo, sin eliminarlo. Se aclarará en cada caso que las capacidades descritas corresponden a la versión comercial completa y que en el prototipo demo dichas características aún no están operativas.

7.1 VISUALIZACIÓN DE RESULTADOS

La aplicación permite visualizar de forma consolidada todos los datos introducidos manualmente por el usuario en la sección de **Medición**. Dado que por el momento no se han implementado cálculos automáticos ni agregaciones complejas, la visualización de resultados se basa en presentar directamente la información proporcionada por el usuario, organizada de manera estructurada y comprensible. Para ello se utilizan tablas HTML y elementos dinámicos que reflejan **en tiempo real** los valores introducidos; es decir, la misma estructura de tablas `contenteditable` o con campos de entrada de la fase de medición se reaprovecha para mostrar un resumen consolidado. En esencia, la plataforma actúa como un contenedor centralizado de datos ESG donde cada métrica introducida queda almacenada temporalmente en la página, permitiendo luego su recopilación visual sin necesidad de recargar la aplicación ni realizar consultas a una base de datos.

El proceso de visualización de resultados se ha diseñado teniendo en cuenta la **trazabilidad** de la información introducida. Cada estándar ESG (definido según los ESRS correspondientes) cuenta con un conjunto de puntos de divulgación o métricas que el

usuario relleno en la etapa de medición. Posteriormente, la sección de **Resultados** presenta esos mismos puntos de divulgación junto con sus respuestas ingresadas, de modo que el usuario pueda revisar de un vistazo toda la información consolidada por estándar o por categoría. La interfaz muestra, por ejemplo, tablas donde cada fila corresponde a una métrica específica (identificada por su código o número de divulgación) y las columnas incluyen el nombre de la métrica, el tipo de dato esperado (por ejemplo, cuantitativo o cualitativo) y la respuesta proporcionada por el usuario. De esta forma, aunque no exista un cálculo automático, la aplicación garantiza que los datos se visualicen de manera ordenada y agrupada lógicamente, sirviendo como **pre-validación** manual de la consistencia de los datos.

Figura 28: Ejemplo de visualización de resultados para el estándar Cambio Climático

Detalles del Estándar ESRS E1				
A continuación, se detallan los puntos de divulgación para el estándar ESRS E1 :				
Número	ESRS	Nombre	Tipo de Datos	Respuesta
1	E1	Divulgación de cómo se tienen en cuenta las consideraciones relacionadas con el clima en la remuneración de los miembros de los órganos administrativos, de gestión y supervisión	Narrativa	La política de retribución de los miembros del Consejo incorpora criterios de sostenibilidad, incluyendo objetivos relacionados con la reducción
2	E1	Porcentaje de la remuneración reconocida que está vinculada a consideraciones relacionadas con el clima	Porcentaje	20 %
3	E1	Explicación de las consideraciones relacionadas con el clima que se tienen en cuenta en la remuneración de los miembros de los órganos administrativos, de gestión y supervisión	Narrativa	El Consejo de Administración revisa periódicamente los riesgos y oportunidades relacionados con el clima. Esta revisión se integra en el
4	E1	Divulgación del plan de transición para la mitigación del cambio climático	Narrativa	La empresa ha definido un plan de transición que contempla la instalación de placas solares en sus instalaciones y el cambio progresivo a proveedores de
5	E1	Explicación de cómo los objetivos son compatibles con la limitación del calentamiento global a un grado y medio en línea con el Acuerdo de París	Narrativa	Se han identificado los principales objetivos climáticos del Acuerdo de París como marco de referencia. Las actividades productivas se alinean con

Fuente: Elaboración propia

En esta tabla de la interfaz web, cada punto de divulgación (identificado por su código, E1-1, E1-2, ...) aparece con su nombre, tipo de dato esperado y la respuesta introducida por el usuario. Esta visualización, generada directamente a partir de los datos de entrada, permite verificar que todas las métricas de E1 han sido correctamente consignadas antes de generar el informe final. La estructura tipo tabla aporta claridad y facilita la revisión por parte del usuario, al presentar lado a lado las preguntas y respuestas de ese estándar.

Cabe destacar que la **estructura dinámica** de la página web hace posible que la visualización de resultados sea inmediata. Gracias al uso de técnicas de Single Page Application simples (mediante secciones ocultas/mostradas y elementos HTML actualizables), los datos introducidos no requieren ser enviados a un servidor para luego mostrarlos, sino que permanecen en el DOM del cliente. Esto implica que en cuanto el usuario navega de la sección de Medición a la sección de Resultados, las mismas variables o elementos HTML contienen ya los valores introducidos y simplemente se exhiben en el formato consolidado. Aunque en la versión actual no hay fórmulas que calculen, por ejemplo, totales, promedios u otros indicadores derivados de los datos, la plataforma deja sentadas las bases para que en futuras iteraciones se puedan incorporar dichas capacidades de cálculo. Por ahora, el enfoque ha sido asegurar la **consistencia visual** y la fidelidad de los datos mostrados respecto a los ingresados manualmente, evitando transformaciones que pudieran introducir errores.

En la práctica, cuando el usuario finaliza la introducción de datos en Medición no aparece ningún panel real de resultados ni botones activos de exportación; la plataforma no despliega el reporte consolidado en esta etapa.

7.2 VISTA PREVIA DEL INFORME

Una vez que el usuario ha completado la introducción de datos en las distintas secciones de Medición y ha podido verificarlos en Resultados, la aplicación ofrece el diseño de una vista previa del informe completo antes de su exportación. Esta vista previa actúa como un borrador consolidado del reporte no financiero ESG, permitiendo al usuario revisar todas las secciones y métricas en un solo documento simulado.

La generación de esta vista previa se realiza de forma dinámica a partir de las tablas y campos ya completados: esencialmente, la aplicación recompone en una sola página todos los datos ingresados, respetando la estructura y el formato del informe esperado (por ejemplo, separando por capítulos o por bloques temáticos ESG). El propósito principal de la vista previa es validar la integridad y coherencia del informe antes de generar la versión

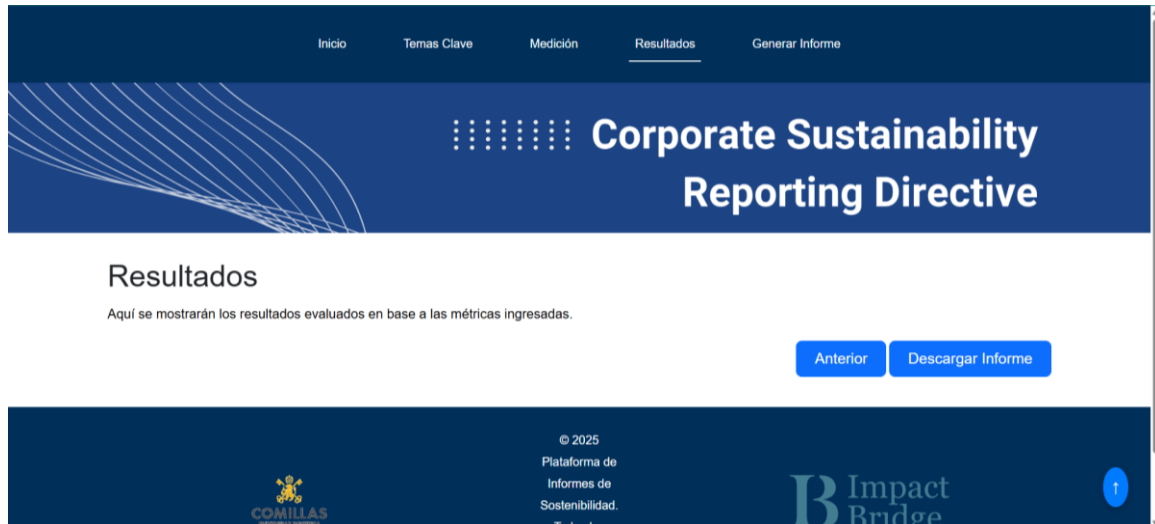
final en PDF o impresa. En esta etapa, el usuario puede detectar fácilmente si falta algún dato, si alguna respuesta es incorrecta o si hay inconsistencias entre secciones.

La interfaz de la vista previa presenta los contenidos de manera estática (ya no editable), simulando cómo se verían en el informe final; no obstante, se proporciona al usuario la posibilidad de regresar a las secciones previas de introducción de datos para corregir cualquier valor erróneo. Esto se logra mediante botones de navegación (botón "Volver a Medición") que aprovechan la naturaleza de aplicación de una sola página para mostrar nuevamente los formularios pertinentes con los datos previamente introducidos. De esta manera, la experiencia es iterativa: el usuario puede alternar entre la vista previa y los formularios de medición hasta quedar satisfecho con la exactitud de todos los datos.

Desde un punto de vista técnico, la construcción de la vista previa del informe implica recorrer los elementos completados (todas las celdas de tablas `contenteditable` o `inputs` de texto de la fase de Medición) y renderizarlos en un formato unificado. Una implementación posible consiste en copiar o referenciar dichos nodos del DOM dentro de una sección específica de Informe que, inicialmente oculta, se muestra al activar la opción de previsualización. Gracias a que los datos residen en el cliente, esta operación se ejecuta sin latencia perceptible. Además, el estilo visual de esta sección está diseñado para imitar un informe formal: por ejemplo, cabeceras con el título de cada estándar o categoría ESG, tablas con bordes y colores de fondo para distinguir encabezados, y un orden lógico que sigue el índice de contenidos definido (ambiental, social, gobernanza entre otros).

En la versión actual del prototipo, no se ha completado la funcionalidad de consolidación automática de tablas ni la exportación final del informe en PDF, tal como se proponía en secciones anteriores. Por el momento, la sección “Resultados” funciona como un marcador de posición (placeholder), mostrando únicamente un mensaje genérico y un botón etiquetado como **“Descargar Informe”**, que enlaza a la vista previa descrita anteriormente. Este botón constituye la funcionalidad principal de cierre del flujo de uso, aunque no activa aún la generación automática del documento, que se contempla como una mejora futura.

Figura 29: Botón descargar informe en página resultados



Fuente: Elaboración propia

En resumen, la vista previa del informe funciona como un paso intermedio de control de calidad: reúne toda la información introducida en un solo lugar y ofrece al usuario la confianza de que, al proceder a la generación definitiva del informe, no habrá sorpresas ni omisiones. Es una funcionalidad fundamental para asegurar la fiabilidad del reporte, especialmente en ausencia de automatizaciones avanzadas, ya que delega en el usuario la validación final de cada dato en contexto.

7.3 GENERACIÓN DEL INFORME (CONCEPTO DE DESARROLLO FUTURO)

Si bien la aplicación en su versión actual se centra en la introducción y visualización de los datos, se ha previsto como **desarrollo futuro** la incorporación de una funcionalidad completa de **generación de informe** final. Esta funcionalidad permitiría compilar todos los datos validados en la vista previa y exportarlos en un formato adecuado para su distribución, típicamente un documento PDF, o bien mediante la impresión directa desde el navegador. El enfoque propuesto para esta generación del informe enfatiza el uso de las

capacidades nativas del navegador web, evitando en la medida de lo posible dependencias de librerías externas.

En concreto, se sugiere implementar un mecanismo que, al activar la generación del informe (por ejemplo, al hacer clic en un botón "Descargar Informe"), invoque la función nativa `window.print()` del navegador. Esta función abre el diálogo de impresión estándar, que en la mayoría de navegadores modernos permite elegir **"Guardar como PDF"** como opción de salida. De esta forma, la aplicación aprovecha directamente la capacidad integrada de los navegadores para crear un PDF del contenido HTML mostrado en pantalla. Esta aproximación presenta varias ventajas: **(1)** simplifica el desarrollo al no requerir bibliotecas JavaScript de conversión (como jsPDF o html2canvas), reduciendo potenciales incompatibilidades o problemas de mantenimiento; **(2)** garantiza que el resultado impreso y/o pdf será una representación fiel de lo que se ve en la vista previa, siempre que se definan correctamente las hojas de estilo para impresión; **(3)** mejora la portabilidad, ya que cualquier usuario con un navegador web puede generar el informe sin instalar software adicional.

No obstante, para lograr un informe profesional mediante `window.print()`, es necesario afinar ciertos detalles técnicos. Por un lado, se debe definir una hoja de estilo `@media print` específica, que adapte la presentación al formato de página (ajustando márgenes, ocultando elementos de navegación o botones innecesarios y paginando el contenido si es muy extenso). Esto asegura que el documento PDF resultante tenga un formato limpio y apropiado para lectura o entrega formal. Por otro lado, es importante gestionar la experiencia de usuario durante la generación: la aplicación podría, por ejemplo, ocultar el botón de descarga justo al imprimir para que no aparezca en el PDF, y luego restaurarlo (tal como se sugiere en implementaciones comunes de impresión dinámica). En versiones futuras, también podría explorarse la integración de esta función con un service worker u opción offline, de modo que el informe pueda generarse aun sin conexión, dado que todo el contenido ya está cargado en el cliente.

Como alternativa a `window.print()`, se reconoce que existen en el ecosistema múltiples librerías especializadas (como las mencionadas jsPDF, pdfmake, Puppeteer para generar PDF del DOM, etc.) que podrían brindar mayor control sobre el formato y la inclusión de gráficos complejos. Sin embargo, en línea con el carácter **ligero y autónomo** de esta aplicación web, la primera iteración de la generación de informes buscará explotar las herramientas nativas. Esto facilita además el cumplimiento de estándares web y evita añadir peso innecesario al proyecto. Sintetizando la idea, la generación del informe se vislumbra como una evolución natural del módulo de resultados: una vez verificada la información, se ofrecerá al usuario un producto final exportable, manteniendo la **filosofía de consultoría KISS** (Keep It Simple, Stupid) en cuanto a implementación técnica, pero asegurando la calidad visual y la integridad de los datos en el documento final.

7.4 PRIVACIDAD Y SEGURIDAD DE LOS DATOS

En la versión actual (demo) de la plataforma, se ha adoptado deliberadamente una postura conservadora en materia de **almacenamiento de datos**, priorizando la privacidad del usuario y la simplicidad del prototipo. Específicamente, la aplicación **no almacena ningún dato de forma persistente** en el navegador; ni el `sessionStorage` ni el `localStorage` son utilizados para guardar la información introducida. Toda la data proporcionada por el usuario reside únicamente en memoria (es decir, en el DOM de la página activa) y se pierde cuando se cierra la sesión o se recarga la página. Si bien esto conlleva una limitación evidente en términos de persistencia (los usuarios deben completar el informe de una sola vez, ya que no pueden recuperar una sesión previa), desde el punto de vista de la **seguridad** y la protección de la información sensible ofrece la ventaja de no dejar rastro local una vez finalizada la sesión. En otras palabras, al no escribir en `localStorage` ni en otras memorias persistentes del navegador, se reduce el riesgo de que terceros con acceso al equipo del usuario puedan extraer información residual del reporte ESG. Este enfoque está en línea con recomendaciones de seguridad que desaconsejan almacenar información sensible en `localStorage`, dado que dicho espacio puede ser susceptible a brechas si código malicioso

logra ejecutarse en el navegador o si alguien tiene acceso físico al sistema de archivos del dispositivo.

No obstante, se reconoce que en futuros desarrollos será necesario equilibrar mejor la **usabilidad** con la privacidad. Para una versión más robusta de la aplicación, se propone implementar un almacenamiento local **temporal** de los datos ingresados, de modo que el usuario no pierda su progreso ante una eventual recarga o cierre accidental de la pestaña. Una opción adecuada podría ser utilizar `sessionStorage` (cuyo alcance dura solo mientras la pestaña esté abierta) para conservar los valores a medida que el usuario navega entre secciones del informe. Esto permitiría, por ejemplo, que, si el usuario pasa de la sección de Medición a Resultados y luego regresa, los campos sigan rellenos con la información previamente introducida, incluso si hubiese alguna recarga accidental de la página durante el proceso. En todo caso, cualquier mecanismo de almacenamiento que se incorpore deberá venir acompañado de **controles de borrado** y gestión del ciclo de vida de los datos: por ejemplo, un botón de "Eliminar datos" que limpie completamente la información guardada una vez generado el informe, o bien el borrado automático del `sessionStorage/localStorage` al cerrar la sesión o después de un tiempo de inactividad.

Otra consideración para versiones futuras es la posible integración de **cifrado o protección** de los datos en reposo local. Si bien `localStorage` y `sessionStorage` son accesibles solo por scripts del mismo dominio, no cifrados por sí mismos, podría añadirse una capa de cifrado de lado cliente para cualquier información particularmente sensible antes de guardarla. Sin embargo, dado que el ámbito de este proyecto es un prototipo de evaluación de potencial, medidas avanzadas como el cifrado local se valorarían en función de la sensibilidad real de los datos manejados y del contexto de despliegue de la herramienta. En escenarios corporativos reales, probablemente se optaría por no depender únicamente del almacenamiento en el navegador, sino por una solución de **backend seguro** donde los datos ESG se almacenen en servidores o bases de datos con las debidas autenticaciones y cifrado.

De manera sucinta, la versión beta de la aplicación sacrifica persistencia por privacidad, asegurándose de que los datos no financieros introducidos para el reporte ESG no queden expuestos más allá de la sesión activa del navegador. Esta decisión, apropiada para demostrar el concepto y evaluar el potencial de la herramienta, se plantea como transitoria. A medida que la plataforma evolucione, deberá incorporar mecanismos de almacenamiento seguro y temporal que mejoren la experiencia del usuario (evitando la pérdida accidental de información) sin comprometer la confidencialidad de los datos sensibles. El equilibrio entre **usabilidad** y **seguridad** será un aspecto clave en la hoja de ruta futura del proyecto, garantizando que la herramienta pueda escalar hacia un entorno productivo cumpliendo con las expectativas de protección de la información de sus usuarios.

Capítulo 8. ADAPTABILIDAD A PYMES E IMPLEMENTACIÓN LIGERA

En este capítulo se analiza la solución desarrollada desde una perspectiva de **diseño centrado en PYMES** y su adecuación al marco normativo actual. Se describe cómo la aplicación web de reporte no financiero ha sido concebida para pequeñas y medianas empresas, con **formularios simplificados** y **navegación intuitiva**, evitando la necesidad de infraestructura tecnológica compleja. Asimismo, se reflexiona sobre la **implementación ligera** del software, basada únicamente en HTML, CSS y JavaScript que permite un despliegue sencillo **sin bases de datos ni servidores** dedicados. A continuación, se presenta un **análisis de la alineación** de las funcionalidades de la plataforma con los requerimientos de la Directiva Europea **CSRD**, especialmente a la luz de su última modificación que **excluye a las PYMES no cotizadas** de la obligatoriedad de reporte. Para ello, se incluye una **tabla comparativa** que relaciona los principales requisitos de la CSRD con las funcionalidades actuales del software, indicando cómo estas podrían **evolucionar** en el futuro para cubrir plenamente los puntos necesarios. Posteriormente, se discuten las posibilidades de **escalabilidad técnica** de la aplicación, proponiendo mejoras como la autenticación de usuarios, integración con bases de datos, generación automática de informes ESG con lógica de **puntuación** de desempeño, exportación de datos en formatos interoperables y la visión de una posible **versión SaaS** (Software as a Service) bajo suscripción. Finalmente, se incorpora una **perspectiva estratégica** retomando la parte inicial del proyecto desarrollada para Impact Bridge, donde se justifica la solución propuesta para pequeñas empresas y se examinan las **barreras de entrada** que enfrentan las PYMES ante soluciones ESG complejas y costosas ya presentes en el mercado.

8.1 DISEÑO DE LA APLICACIÓN CENTRADO EN LAS PYMES

El software fue diseñado desde sus inicios pensando en las necesidades y limitaciones de las **pequeñas y medianas empresas (PYMES)** en materia de reporte de sostenibilidad. A diferencia de las grandes corporaciones, que suelen contar con departamentos especializados en **Responsabilidad Social Corporativa** o recursos para contratar consultorías, las PYMES típicamente carecen de personal dedicado exclusivamente a ESG. Por ello, la aplicación adopta una interfaz **simplificada y auto-explicativa**, con formularios claros y una navegación paso a paso que no requiere conocimientos técnicos avanzados ni experiencia previa en marcos de reporte. En la página de inicio se introducen los conceptos esenciales (por ejemplo, qué es la CSRD y por qué es importante) en un lenguaje comprensible, preparando al usuario para el proceso de recopilación de datos. Seguidamente, la plataforma organiza el llenado de información en **módulos temáticos** que siguen la lógica de los estándares ESG aplicables (ambiental, social y gobernanza), de modo que el usuario puede **progresar de forma estructurada** a través de secciones lógicas en lugar de enfrentarse a un formulario monolítico extenso.

La **navegación** ha sido diseñada para ser intuitiva incluso para usuarios con escasa experiencia en herramientas digitales avanzadas. Un menú principal permite alternar entre secciones (Introducción, Temas Clave, Medición, Resultados e Informe) de forma **clara y consistente**, funcionando como una guía del proceso de reporte. Esta estructura modular facilita que las PYMES aborden el reporte **por etapas**: primero comprendiendo *qué* deben reportar (sección de *Temas Clave*), luego *cómo medir o ingresar* sus datos (sección de *Medición*), para finalmente revisar los *resultados* agregados y **generar el informe** final. Cada formulario parcial presenta solo un subconjunto manejable de preguntas o indicadores, evitando abrumar al usuario con demasiada información en pantalla. Adicionalmente, se emplean componentes web nativos (como tablas HTML, casillas de texto y selectores) conocidos por cualquier usuario promedio, garantizando que la interacción sea **familiar** y sin necesidad de capacitación especial.

Otro aspecto importante del diseño centrado en PYMEs es la **ausencia de requerimientos de infraestructura interna**. La empresa usuaria no necesita disponer de un servidor propio, ni instalar software pesado o licencias costosas; basta con un navegador web moderno para utilizar la aplicación. Esta filosofía zero-footprint (cero infraestructuras) responde a la realidad de muchas PYMEs, que suelen carecer de departamentos de TI robustos. En suma, la aplicación **minimiza las barreras de uso**: es accesible, autosuficiente y reduce la complejidad técnica al mínimo, permitiendo que pequeñas organizaciones puedan iniciarse en el **reporte ESG** sin enfrentarse a la curva de aprendizaje ni a los costes asociados a soluciones empresariales de mayor envergadura.

8.2 IMPLEMENTACIÓN LIGERA Y DESPLIEGUE SIMPLIFICADO

Un factor clave que hace a esta plataforma atractiva para las PYMEs es su **implementación ligera** en términos tecnológicos. Toda la solución está construida con tecnologías web estándar del lado del cliente: **HTML, CSS y JavaScript**. Esto significa que la aplicación es esencialmente un *sitio web estático*, sin componentes de servidor ni dependencias de software específicas más allá de un navegador. La elección de una arquitectura frontend-only conlleva múltiples beneficios para el despliegue y la mantenibilidad:

- **Sin base de datos ni servidor dedicado:** Todos los datos introducidos por el usuario se manejan en el contexto local (por ejemplo, utilizando almacenamiento en el navegador como LocalStorage o SessionStorage). Al no requerir una base de datos externa, se elimina la necesidad de configurar y mantener motores de almacenamiento o servidores backend. Igualmente, la lógica de negocio y de presentación reside completamente en el código JavaScript ejecutado en el cliente, prescindiendo de un servidor de aplicación. Esto reduce drásticamente la complejidad de despliegue: la aplicación puede **ejecutarse localmente** abriendo el archivo HTML en un navegador, o bien servirse desde cualquier hosting básico (incluso gratuito) sin más que subir los ficheros estáticos.

- **Fácil despliegue y portabilidad:** La ausencia de componentes pesados permite que la solución sea muy portátil. Una PYME puede desplegar la herramienta en su propio sitio web corporativo como un conjunto de páginas HTML, o incluso utilizarla de manera interna simplemente abriendo el archivo en equipos de la oficina. No se requieren procesos de instalación complicados, ni entornos específicos (la aplicación es independiente del sistema operativo y funciona en cualquier navegador moderno). Esto también implica que las actualizaciones o correcciones son tan sencillas como reemplazar los archivos HTML/JS, sin downtime apreciable ni necesidad de migraciones de datos.
- **Bajo consumo de recursos:** Al no haber procesamiento en servidor, el rendimiento depende principalmente del navegador del cliente. Dado que las operaciones de la aplicación (gestionar formularios, cálculos básicos de agregación, navegación entre secciones) son poco demandantes, una máquina estándar es suficiente. Esto resulta idóneo para PYMEs que quizá no cuentan con hardware de altas prestaciones ni desean contratar servicios en la nube costosos. La aplicación ha demostrado ser ágil en entornos de prueba, con **tiempos de carga mínimos** y respuesta inmediata a las interacciones del usuario, gracias a su tamaño reducido y la ausencia de llamadas a servidores remotos.
- **Mantenimiento simplificado:** Con todo el código en tecnologías ampliamente conocidas (HTML/CSS/JS) y sin frameworks complejos, el mantenimiento o personalización de la herramienta resulta más asequible. Un desarrollador web generalista puede entender el funcionamiento y adaptar textos o estilos si la empresa lo requiere, sin tener que lidiar con arquitecturas de software empresariales. Esto es coherente con el perfil de las PYMEs, que valoran soluciones sencillas que no las hagan depender de personal altamente especializado para su adaptación o mejora.

Esta implementación ligera, no obstante, conlleva **algunas limitaciones inherentes**. Por ejemplo, al no existir una base de datos, la persistencia de la información es limitada a cada sesión de uso (a menos que se exporten manualmente los resultados). Del mismo modo, la

ausencia de un backend impide, en la versión actual, la colaboración multiusuario en línea o el almacenamiento centralizado de múltiples reportes históricos. Estas limitaciones fueron aceptadas explícitamente en el diseño inicial a cambio de maximizar la **facilidad de uso e implantación**, considerando el alcance acotado del proyecto y las prioridades de las PYMEs objetivo. En cualquier caso, la decisión de una arquitectura ligera ha demostrado ser acertada para efectos de **demostración y validación**: cualquier PYME interesada puede probar la herramienta de inmediato, sin inversión en infraestructura, lo que reduce la **brecha** entre la intención de reportar ESG y la acción efectiva de comenzar a hacerlo.

8.3 ALINEACIÓN CON LOS REQUISITOS DE LA DIRECTIVA CSRD

La Directiva CSRD establece un nuevo estándar en la Unión Europea para la transparencia corporativa en materia de sostenibilidad, ampliando significativamente el número de empresas obligadas a reportar información ESG. En su forma aprobada, la CSRD **alcanza a todas las grandes empresas y a las PYMEs cotizadas** en mercados regulados (exceptuando microempresas), escalonando su entrada en vigor entre 2024 y 2026. **Las pequeñas y medianas empresas no cotizadas quedan fuera del mandato legal**, si bien la normativa contempla un estándar simplificado y voluntario para ellas. De hecho, el organismo EFRAG publicó en diciembre de 2023 un **Estándar Voluntario de Reporte de Sostenibilidad para PYMEs no cotizadas (VSME)**, como parte del SME Relief Package, reconociendo la importancia de facilitar a estas empresas un marco adaptado a su escala. Este estándar voluntario busca convertirse en referencia de mercado para que las PYMEs **respondan de manera eficiente** a las solicitudes de información sostenible por parte de sus clientes corporativos o entidades financieras sujetas a la CSRD, estandarizando cuestionarios y **reduciendo la carga burocrática** para ellas. En otras palabras, aunque una PYME no cotizada no tenga obligación legal directa, la **“efecto cascada”** de la CSRD hará que sus grandes clientes o bancos requieran datos ESG, por lo que disponer de una herramienta para recopilar y reportar dicha información resulta igualmente valioso.

En este contexto, la aplicación desarrollada se alinea de forma notable con los **requerimientos de la CSRD y los estándares ESRS** asociados. Desde el punto de vista del **contenido**, la plataforma fue diseñada tomando como referencia explícita los **ESRS** emitidos por EFRAG, que concretan las materias a divulgar bajo la CSRD. En la sección de *Temas Clave* de la aplicación, se listan todos los **temas ESG relevantes** definidos por los ESRS, abarcando el espectro completo de divulgación: cinco dominios ambientales (E1 a E5, que incluyen clima, contaminación, agua, biodiversidad y economía circular), cuatro dominios sociales (S1 a S4, relativos a condiciones laborales internas, trabajadores en la cadena de valor, comunidades locales afectadas y consumidores) y el dominio de gobernanza (G1, que cubre la conducta empresarial ética). Esto asegura que una empresa que utilice la herramienta está considerando **todos los ámbitos** que la CSRD espera que se informen en un reporte de sostenibilidad. Además, la sección de *Medición* profundiza en los **requisitos específicos de divulgación** para cada estándar, presentando los indicadores y datos concretos que la empresa debe proporcionar. Por ejemplo, para el estándar E1 (Cambio Climático) se solicitan las métricas de emisiones de gases de efecto invernadero, desglosadas por alcances, así como la descripción de riesgos y oportunidades climáticos y las estrategias de mitigación adoptadas. Esta estructura refleja directamente las exigencias mínimas de divulgación (minimum disclosure requirements, MDR) que la CSRD impone, garantizando una **correspondencia directa** entre lo que la empresa ingresa en la aplicación y lo que debería figurar en su informe de sostenibilidad conforme a la ley.

Otro pilar de la CSRD es el principio de **doble materialidad**, que obliga a las empresas a reportar no solo cómo los factores ESG pueden afectarles financieramente, sino también cómo las actividades de la empresa impactan al medio ambiente y la sociedad. La herramienta incorpora este enfoque de manera transversal: muchos de los campos solicitados en los formularios ESRS requieren a la PYME describir tanto sus **riesgos y oportunidades** en materia, por ejemplo, climática (cómo el cambio climático puede influir en su modelo de negocio), como sus **impactos** ambientales (cómo sus operaciones contribuyen a las emisiones de GEI). Si bien la versión actual de la aplicación no realiza un análisis automatizado de materialidad (es decir, no determina por sí misma qué temas

son más significativos para la empresa), sí facilita que el usuario considere todos los temas posibles y reporte información en cada uno. A futuro, se podría integrar un módulo de **evaluación de materialidad** que asista a la PYME en la identificación de sus asuntos ESG más relevantes (por ejemplo, mediante un breve cuestionario de relevancia), permitiendo concentrar esfuerzos en ellos. Esto sería coherente con la aproximación de **proporcionalidad** que subyace al estándar voluntario para PYMES: enfocarse en lo esencial sin dejar de cubrir los requerimientos fundamentales.

En la siguiente **Tabla 1** se presenta una comparación entre los principales **requerimientos de la CSRD** y las funcionalidades actuales de la plataforma, indicando asimismo las posibles **evoluciones futuras** para asegurar una alineación completa con la normativa:

Tabla 1. Comparativa entre requisitos de la Directiva CSRD y la funcionalidad de la aplicación desarrollada, indicando mejoras futuras para reforzar la alineación.

Requisito CSRD	Funcionalidad actual del software	Evolución futura propuesta
Alcance de empresas obligadas – Grandes empresas (desde 2024) y PYMES cotizadas (desde 2028); PYMES no cotizadas excluidas o voluntarias.	La aplicación está dirigida a PYMES y puede ser usada voluntariamente por cualquier empresa. Actualmente cubre todo el espectro ESG exigido, resultando útil para PYMES cotizadas que deban cumplir en 2026-27, y para no cotizadas que quieran anticiparse o responder a requerimientos de clientes.	Incorporar un modo simplificado alineado con el estándar voluntario para PYMES no cotizadas (VSM), reduciendo la carga para las más pequeñas. Posibilidad de configurar la herramienta según el tipo/tamaño de empresa para personalizar el alcance de los datos solicitados.
Temáticas ESG a reportar – Información sobre factores medioambientales, sociales y de gobernanza, según estándares ESRS (E1-E5, S1-S4, G1, etc.).	La plataforma organiza los formularios por cada estándar ESRS, asegurando que se cubren todos los temas: clima, contaminación, recursos hídricos, biodiversidad, economía circular, empleados, cadena de valor, comunidades, consumidores y conducta empresarial. Cada tema incluye campos para las divulgaciones requeridas (políticas, métricas, objetivos, etc.).	Actualización continua conforme evolucionen los estándares: p. ej., integrar futuros ESRS sectoriales o modificaciones a estándares existentes. A largo plazo, podría añadirse un módulo de sector para incluir requisitos específicos de ciertas industrias si la CSRD desarrolla estándares sectoriales.
Principio de doble materialidad – Reportar tanto impactos de la empresa en ESG como efectos de ESG sobre la empresa (riesgos y oportunidades).	Los formularios actuales recaban descripciones cualitativas y métricas que cubren ambos aspectos (Impactos y riesgos). Por ejemplo, se solicita describir cómo las operaciones afectan al clima y cómo el cambio climático afecta el negocio. No obstante, se deja al usuario la identificación de qué es material; la aplicación no distingue automáticamente la relevancia de cada tema.	Incluir herramientas de materialidad asistida: por ejemplo, matrices de materialidad interactivas donde la PYME pueda evaluar la relevancia interna y externa de cada tema ESG. La aplicación podría entonces filtrar o destacar los temas más materiales y generar automáticamente justificaciones de omisión para temas de baja materialidad, cumpliendo con el principio de proporcionalidad para PYMES.
Formato de informe y accesibilidad digital – El informe de sostenibilidad debe integrarse en el informe de gestión anual y publicarse en formato electrónico, etiquetado para su fácil acceso y comparabilidad.	La herramienta genera un informe en HTML (y opcionalmente imprimible/PDF) que la empresa puede incorporar a su memoria anual o sitio web. Este informe digital facilita la difusión y consulta por stakeholders. Sin embargo, actualmente carece de etiquetado XBRL u otro formato de datos estructurados exigido para el registro oficial electrónico.	Desarrollar funciones de exportación en formatos estándar, por ejemplo: generación de un informe en PDF con todos los contenidos para adjuntar al informe anual; exportación a XBRL/XBRL conforme a la taxonomía ESRS una vez esté disponible, para cumplir con la presentación digital oficial; y servicios web o APIs para compartir datos con plataformas externas. Esto asegurará la interoperabilidad y cumplimiento de los formatos regulatorios.
Verificación y garantía – La información reportada deberá someterse a verificación por un auditor independiente (garantía limitada inicialmente, razonable en el futuro).	Actualmente, el software no incorpora funcionalidades de auditoría o validación externa de los datos ingresados; se enfoca en la auto-declaración por parte de la empresa. Los resultados dependen de la veracidad y calidad de los datos que introduzca el usuario, sin un proceso interno de comprobación más allá de controles de formato básicos en campos numéricos.	Integrar opciones para soporte de auditoría: por ejemplo, un modo de usuario verificador que pueda acceder de forma segura a los datos ingresados y adjuntar comentarios o documentos de respaldo. También se podría implementar la generación de pistas de auditoría (logs de cambios, marcas de tiempo, fuentes de datos) que faciliten a un tercero la revisión. Aunque la verificación final seguirá siendo externa, la herramienta puede ayudar proporcionando datos bien estructurados y evidencias que agilicen la labor del auditor.

Fuente: Elaboración propia

Como se aprecia en la tabla, la aplicación cubre en su estado actual gran parte de las necesidades de cumplimiento con la CSRD para una PYME, especialmente en lo referente al **alcance temático y contenido ESG** a reportar. La alineación conceptual es sólida: la estructura modular por estándares ESRS garantiza que no se omitan tópicos importantes, y la información recopilada sería la base adecuada para un informe conforme a la ley. No obstante, existen áreas donde **serán necesarias evoluciones** para alcanzar una conformidad

plena a nivel formal. Particularmente, la cuestión del **formato digital oficial** (etiquetado de datos) y la integración de mecanismos que faciliten la **verificación externa** son aspectos pendientes. Dado que las PYMEs cotizadas deberán empezar a reportar bajo CSRD a partir del ejercicio 2026 (informando en 2027), y considerando que la Comisión Europea prevé estándares diferenciados para PYMEs con proporcionalidad antes de esa fecha, la aplicación está bien situada como punto de partida y puede adaptarse a los detalles finales de la normativa. Además, para las PYMEs no obligadas, la herramienta sirve como guía voluntaria para **mejorar su transparencia** y preparar información ESG de forma estructurada, algo cada vez más valorado por inversores, bancos y socios comerciales incluso en ausencia de obligación legal. De hecho, la adopción voluntaria de reportes ESG puede redundar en beneficios como un **mejor acceso a financiación** y una **mayor competitividad** en el mercado, alineándose con los objetivos finales de la regulación de impulsar la sostenibilidad corporativa en todos los niveles.

8.4 ESCALABILIDAD TÉCNICA Y MEJORAS FUTURAS PROPUESTAS

Si bien la solución actual cumple su cometido en entornos controlados y para un solo usuario a la vez, se han identificado varias líneas de mejora para dotarla de mayor **escalabilidad técnica** y funcional, especialmente pensando en una adopción más amplia o comercial (como producto ofrecido a múltiples PYMEs). A continuación, se enumeran propuestas concretas de evolución futura:

- **Autenticación de usuarios y control de acceso:** Incorporar un sistema de **login** seguro que permita a múltiples usuarios (por ejemplo, distintos clientes PYMEs) acceder a la aplicación de forma aislada. Cada empresa podría tener su cuenta protegida por credenciales, asegurando la confidencialidad de sus datos. Esta autenticación sentaría las bases para ofrecer la herramienta en modo **SaaS multiusuario**, donde cada PYME gestiona su información en la nube de forma privada.

- **Integración con base de datos y persistencia en la nube:** Migrar desde el almacenamiento exclusivamente local en el navegador hacia una **base de datos** central (SQL, NoSQL o un servicio cloud) permitiría **persistir** los datos de reporte de cada usuario a lo largo del tiempo. Esto habilitaría funcionalidades como guardar avances, editar informes en múltiples sesiones, llevar histórico de años anteriores y comparar progreso. La base de datos sería el pilar para escalar a cientos de usuarios simultáneos, garantizando además integridad y respaldos de la información.
- **Generación automática de informes ESG con lógica de puntuación:** Actualmente la aplicación recopila datos y los muestra tal cual en un informe. Una mejora sería añadir una capa de **análisis automatizado**, donde el software interprete las respuestas y calcule **indicadores compuestos o puntuaciones ESG**. Por ejemplo, podría asignarse una puntuación de desempeño en cada dimensión (ambiental, social, gobernanza) en base a las métricas introducidas y compararlas con benchmarks sectoriales. También se podrían generar visualizaciones (gráficos de barras, radares ESG) para resumir el nivel de madurez de la empresa en sostenibilidad. Esta lógica de puntuación brindaría un **feedback inmediato** a la PYME sobre dónde está bien posicionada y en qué áreas necesita mejora, agregando valor más allá del mero cumplimiento.
- **Exportación de resultados en formatos interoperables:** Para maximizar la utilidad de los datos recopilados, se propone implementar opciones de **exportación** del informe en diversos formatos. Por un lado, un **PDF** formateado profesionalmente facilitaría compartir el reporte con terceros o integrarlo como anexo en memorias anuales. Por otro lado, exportaciones en **Excel/CSV** podrían permitir a la empresa reutilizar los datos internamente o cargarlos en otros sistemas. Más importante aún, como ya se mencionó, sería estratégico soportar exportación a **formato XBRL** (o el requerido por el regulador) para presentación oficial, lo que haría a la herramienta plenamente compatible con las futuras exigencias de **reporting electrónico** de la CSRD.

- **Implementación de una plataforma SaaS bajo suscripción:** Todas las mejoras anteriores convergen en la posibilidad de transformar la solución en un servicio web completo. Bajo este modelo Software as a Service, una organización (por ejemplo, una startup derivada del proyecto) alojaría la aplicación en la nube y la ofrecería a múltiples PYMEs mediante suscripciones. Las ventajas serían múltiples: las PYMEs evitarían cualquier instalación local, siempre accederían a la última versión actualizada del software, y podrían beneficiarse de funcionalidades avanzadas (como copias de seguridad automáticas, soporte técnico, mejoras continuas) incluidas en la suscripción. Para el proveedor, este modelo permitiría escalar el impacto del producto y sostener su mantenimiento mediante ingresos recurrentes. Desde el punto de vista técnico, implicaría fortalecer la **seguridad** (certificados SSL, protección de datos sensibles, cumplimiento de GDPR), optimizar la aplicación para servidores (posiblemente adaptando parte de la lógica a backend para heavy lifting si la base de usuarios crece mucho) y garantizar la **escalabilidad horizontal** (capacidad de atender concurrentemente a muchos usuarios, por ejemplo, mediante balanceo de carga o contenedores en la nube).
- **Otras mejoras diversas:** Adicionalmente, se pueden considerar mejoras como la internacionalización del interfaz (traducción a inglés u otros idiomas, dado que la CSRD aplica en toda la UE), la incorporación de **ayudas contextuales** (ejemplo: al pasar el cursor por un indicador, mostrar la definición o consejos para rellenarlo, tomando definiciones de ESRS), e incluso integrar módulos de **formación online** para PYMEs dentro de la aplicación, que expliquen mejores prácticas ESG. Estas funciones apoyarían aún más el objetivo de acompañar a las pequeñas empresas en su viaje hacia la sostenibilidad, más allá de la simple recolección de datos.

En conjunto, las mejoras propuestas buscan **equipar a la aplicación con características propias de un sistema robusto**, sin perder de vista la filosofía de simplicidad y accesibilidad. Ciertamente, cada añadido deberá evaluarse cuidadosamente para mantener un balance entre funcionalidades avanzadas y usabilidad para el usuario no experto. No obstante, dado el interés creciente en soluciones digitales para el cumplimiento ESG, existe

una oportunidad para que una versión evolucionada de esta plataforma se posicione competitivamente, llenando un nicho entre las herramientas excesivamente básicas (hojas de cálculo manuales) y los softwares empresariales integrales pero costosos.

8.5 PERSPECTIVA ESTRATÉGICA: JUSTIFICACIÓN DE LA SOLUCIÓN PARA PYMES Y BARRERAS DE ENTRADA EN EL MERCADO

Desde una óptica estratégica de negocio, la solución propuesta en este proyecto responde a una **necesidad emergente** en el segmento de las PYMEs. Si bien las grandes corporaciones llevan años realizando reportes de sostenibilidad apoyadas por consultoras especializadas y software de alta gama, las pequeñas empresas históricamente han quedado al margen por falta de obligaciones normativas y por los **altos costes y complejidad** que conlleva implementar prácticas ESG formales. Con la llegada de la CSRD y, en general, con la creciente presión de inversores, clientes y la sociedad por la transparencia en sostenibilidad, las PYMEs se encuentran en una encrucijada: **reconocen la importancia** de integrar criterios ESG, pero **perciben barreras significativas** para hacerlo. De acuerdo con un informe reciente de Finresp y Cepyme (2024), más del **80% de las PYMEs** identifica barreras para la integración de la ESG en sus operaciones, citando principalmente los **altos costes** asociados y la falta de recursos debido a su tamaño limitado. En palabras del presidente de Cepyme, actualmente se ha convertido la sostenibilidad en una "sobrerregulación" con "exceso de cargas" que suponen un **gran coste para las pequeñas** empresas. Este sentir resume la realidad de muchas PYMEs: aunque quisieran avanzar en sostenibilidad, temen verse abrumadas por requisitos complejos, consultorías onerosas o software que excede sus capacidades financieras y técnicas.

Frente a estas barreras de entrada, la aplicación desarrollada ofrece una **propuesta de valor ajustada a las PYMEs**. En primer lugar, reduce drásticamente el **coste económico** de iniciarse en el reporte ESG, al estar construida con tecnología abierta y potencialmente distribuable de forma gratuita o a muy bajo coste comparado con las herramientas empresariales del mercado. Una PYME típica evitaría así desembolsos fuertes en licencias

o consultores externos para cumplir con sus reportes no financieros. En segundo lugar, la solución minimiza la **complejidad técnica**: como se detalló, no requiere instalaciones complicadas ni personal de TI dedicado, algo crucial dado que más del **60% de las empresas** (particularmente pymes) carece de expertos internos en medición de sostenibilidad y carbono. La interfaz simple e intuitiva actúa casi como un *asesor virtual*, guiando a la empresa paso a paso por lo que necesita reportar, con explicaciones incorporadas de los términos ESG. Esto **democratiza** el acceso al reporting: incluso organizaciones sin experiencia previa pueden generar su primer informe ESG con cierto rigor siguiendo el flujo de la aplicación. [Staniforth, J. (2025, May 13)].

Otro elemento estratégico es el **enfoque en la proporción y la relevancia**. A diferencia de las soluciones diseñadas para grandes compañías que suelen ser exhaustivas pero engorrosas para una estructura pequeña, la plataforma puede adaptarse para que una PYME reporte de manera **proporcionada a su realidad**, centrándose en los indicadores esenciales. Esto está alineado con la filosofía regulatoria que se vislumbra: estándares voluntarios y simplificados para PYMEs, y la opción de opt-out temporal que la CSRD ofrece a las cotizadas más pequeñas para darles más tiempo. La solución permite a una PYME **aprender haciendo**. En el proceso de llenar los formularios, la empresa va comprendiendo mejor sus propios impactos y riesgos ESG, generando un know-how interno valioso. Esta curva de aprendizaje suavizada es un diferencial importante frente a otras alternativas donde la PYME tendría que estudiar por separado extensos manuales de reporting o contratar capacitación especializada.

Desde el punto de vista de mercado, existe actualmente un **espacio poco cubierto** entre las herramientas extremadamente básicas (por ejemplo, plantillas de Excel genéricas para informes) y las plataformas integrales de sostenibilidad orientadas a grandes corporaciones (que incluyen funcionalidades como inteligencia de datos, cumplimiento de múltiples marcos GRI, SASB, TCFD y que suelen implicar implementaciones largas y costosas). Este espacio es precisamente el que ocupan las PYMEs que quieren (o necesitan) reportar ESG de **forma seria pero sencilla**. La aplicación desarrollada apunta a ese nicho con una solución plug-and-play, lista para usar, y enfocada en el **estándar europeo ESRS**, lo que

le da relevancia inmediata en el contexto post-CSR. Cabe destacar que, aunque la obligatoriedad legal para muchas PYMEs sea limitada, hay incentivos de mercado en juego: según el citado informe, tres de cada cuatro pequeñas y medianas empresas encuestadas ven **beneficioso** realizar esfuerzos en materia ESG, principalmente por la mejora en la **reputación** corporativa que ello conlleva. También se espera que las empresas sostenibles ganen atractivo para ciertos clientes y cadenas de suministro, e incluso obtengan acceso preferente a financiación (muchos bancos comienzan a evaluar riesgos ESG de sus acreditados). Por tanto, una herramienta que facilite a las PYMEs comunicar su sostenibilidad puede convertirse en una aliada estratégica para ellas, más allá del cumplimiento normativo.

De manera sintética, desde la perspectiva estratégica el desarrollo de esta aplicación web se justifica plenamente por la convergencia de **necesidad** (nuevos requerimientos e interés de las PYMEs por ESG) y **oportunidad** (ausencia de soluciones accesibles adaptadas a este segmento). Al atacar las barreras de entrada de coste y complejidad con una propuesta tecnológica simple y asequible, se reducen las **fricciones** que impedían a muchas pequeñas empresas sumarse al reporte de sostenibilidad. De esta forma, el proyecto no solo entrega un prototipo funcional, sino que sienta las bases para una solución escalable que, con las mejoras planteadas, podría contribuir a que un mayor número de PYMEs transiten hacia prácticas de transparencia ESG. Esto tiene un impacto potencial significativo, dado que las PYMEs constituyen el **tejido empresarial mayoritario** en la economía: lograr que adopten herramientas de reporte sostenible, aunque sea de modo voluntario, amplifica la transparencia global y extiende la cultura de sostenibilidad corporativa a niveles antes poco alcanzados.

Capítulo 9. VALIDACIÓN DEL PROTOTIPO CON USUARIOS

9.1 VALIDACIÓN DEL PROTOTIPO EN ENTORNOS EMPRESARIALES

A continuación, se presenta la **validación del prototipo de la plataforma web de informes ESG** mediante **pruebas piloto** realizadas con tres empresas españolas de distintos perfiles. El objetivo de estas pruebas es simular un escenario real de uso, evaluando la **aplicabilidad, usabilidad y robustez** del sistema en situaciones concretas. Dada la creciente importancia del reporting de sostenibilidad, especialmente ante la inminente entrada en vigor de la Directiva CSRD, que obligará a miles de empresas (incluyendo PYMEs cotizadas) a reportar información ESG a partir de 2026-2027 resulta fundamental comprobar cómo el prototipo asiste a diferentes tipos de organizaciones en este desafío. A continuación, se detallan **tres casos de validación** representativos: (1) una pequeña firma industrial, (2) una startup orientada a la sostenibilidad, y (3) una empresa manufacturera mediana. En cada caso se describen los **perfiles de usuario**, las **tareas realizadas en la plataforma**, el **entorno de prueba** (simulado como una sesión piloto real) y la **retroalimentación cualitativa** recopilada.

Figura 30: Perfiles de usuarios reales en la prueba del prototipo



Fuente: Elaboración propia

9.2 VALIDACIÓN EN UNA PEQUEÑA EMPRESA INDUSTRIAL (GRUPO INDUSTRIAL GARCÍA)

Perfil de la empresa y usuarios: El primer piloto se llevó a cabo en una **pequeña empresa industrial** del sector metalúrgico, perteneciente al Grupo Industrial García. Se trata de una firma con unos 50 empleados dedicada a la fabricación de componentes metálicos, la cual hasta el momento no contaba con experiencia formal en elaboración de informes ESG. La empresa no está obligada por ley a reportar sostenibilidad (no cotiza en bolsa), pero **desea anticiparse a las prácticas de RSC** y atender requisitos de clientes más grandes que solicitan compliance ambiental y social en su cadena de suministro. Participó en la sesión piloto el **Gerente de Calidad y Medio Ambiente**, quien en esta PYME asume también las funciones de sostenibilidad corporativa. Este usuario posee un perfil técnico (ingeniero industrial) con conocimientos en normativa ISO 14001, pero **poca familiaridad con estándares ESG integrales** como los ESRS europeos.

Entorno de prueba: La validación se realizó de manera presencial en las oficinas de la empresa, en una sesión de aproximadamente dos horas. El prototipo se desplegó en un ordenador portátil proporcionado por el equipo del proyecto. Antes de iniciar, se brindó al participante una breve introducción sobre el objetivo de la plataforma y se recordó el contexto normativo (CSRD/ESRS) que da lugar a su desarrollo. A partir de allí, el usuario interactuó libremente con la aplicación, asumiendo las tareas típicas de la preparación de un informe de sostenibilidad. El investigador adoptó un rol observador, tomando nota de comentarios y cronometrando ciertas actividades, pero sin intervenir salvo ante dudas explícitas del usuario. Se simuló así una **situación real de auto-uso**, donde el **gerente de la PYME exploró las funcionalidades** de la plataforma tal como lo haría si la hubiese adoptado internamente.

Tareas realizadas: El **flujo de trabajo seguido** por el usuario de la pequeña industrial fue lineal, siguiendo la estructura propuesta por la plataforma. En primer lugar, navegó por la sección “**Temas Clave del Informe**” para familiarizarse con los **estándares ESG incluidos** y el alcance del reporte. Allí observó que el sistema listaba los distintos **estándares ESRS aplicables** (generales, medioambientales, sociales, etc.), junto con una breve descripción de cada uno y un indicador de estado (completado/pendiente). Este listado le resultó útil para tener una **visión general de las obligaciones de divulgación**. A continuación, procedió a la sección “**Medición**”, donde comenzó a **introducir datos y respuestas** a los puntos de divulgación específicos. En particular, el usuario decidió concentrarse en el estándar ambiental *ESRS EI* (Cambio Climático), dado que su empresa tiene datos relacionados con emisiones y consumo energético. **Completó campos cualitativos y cuantitativos** correspondientes a dicho estándar, tales como la descripción de cómo la dirección considera el cambio climático en su estrategia, y métricas como el porcentaje de remuneración ligado a criterios climáticos, entre otros.

Figura 31: Interfaz con datos personales de los puntos del estándar ESRS E1

Detalles del Estándar ESRS E1

A continuación, se detallan los puntos de divulgación para el estándar **ESRS E1**:

Número	ESRS	Nombre	Tipo de Datos	Respuesta
1	E1	Divulgación de cómo se tienen en cuenta las consideraciones relacionadas con el clima en la remuneración de los miembros de los órganos administrativos, de gestión y supervisión	Narrativa	El comité de dirección evalúa los riesgos climáticos en sus reuniones trimestrales. Se tiene en cuenta la exposición al clima al fijar los bonus
2	E1	Porcentaje de la remuneración reconocida que está vinculada a consideraciones relacionadas con el clima	Porcentaje	20 %
3	E1	Explicación de las consideraciones relacionadas con el clima que se tienen en cuenta en la remuneración de los miembros de los órganos administrativos, de gestión y supervisión	Narrativa	Se ha considerado el cambio climático como un criterio transversal en la toma de decisiones. La dirección ha tenido en cuenta los objetivos climáticos en
4	E1	Divulgación del plan de transición para la mitigación del cambio climático	Narrativa	Actualmente no contamos con un plan de transición climática formal. Sin embargo, estamos explorando opciones para reducir nuestras emisiones
5	E1	Explicación de cómo los objetivos son compatibles con la limitación del calentamiento global a un grado y medio en línea con el Acuerdo de París	Narrativa	Nos guiamos por el Acuerdo de París y buscamos contribuir a la meta de 1.5 °C, especialmente optimizando el consumo energético en nuestros procesos
6	E1	Divulgación de palancas de descarbonización y acciones clave	Narrativa	Nuestra principal palanca de descarbonización es el cambio a proveedores eléctricos con garantía de origen 100% renovable. También estamos
7	E1	Divulgación de gastos operativos significativos (OpEx) y/o gastos de capital (CapEx) necesarios para la implementación del plan de acción	Narrativa	Estimamos una inversión de aproximadamente 5.000 € en eficiencia energética en 2024, principalmente para modernizar sistemas de iluminación y
8	E1	Recursos financieros asignados al plan de acción (OpEx)	Monetario	5000 €
9	E1	Recursos financieros asignados al plan de acción (CapEx)	Monetario	0 €
10	E1	Explicación de posibles emisiones de GEI bloqueadas por activos y productos clave y cómo estas emisiones bloqueadas podrían comprometer el logro de los objetivos de reducción de emisiones de GEI y generar riesgos de transición	Narrativa	No se han identificado emisiones de GEI bloqueadas en activos o procesos críticos que comprometan nuestros objetivos climáticos.
11	E1	Explicación de cualquier objetivo o plan (CapEx, planes de CapEx, OpEx) para alinear las actividades económicas (ingresos, CapEx, OpEx) con los criterios establecidos en el Reglamento Delegado de la Comisión 2021/2139	Narrativa	Aunque no disponemos de un plan formal, estamos alineando nuestras inversiones en equipamiento con el objetivo de reducir consumo energético. Aplicamos
12	E1	CapEx significativo para actividades económicas relacionadas con el carbón	Monetario	0 €
13	E1	CapEx significativo para actividades económicas relacionadas con el petróleo	Monetario	0 €
14	E1	CapEx significativo para actividades económicas relacionadas con el gas	Monetario	0 €
15	E1	La empresa está excluida de los índices de referencia alineados con el Acuerdo de París de la UE	Seminarrativa	La empresa no forma parte de índices alineados con el Acuerdo de París. No cotiza en bolsa ni tiene compromisos

Fuente: Elaboración propia

Esta captura de pantalla ilustra el tipo de campos (textuales y numéricos) que el usuario de la empresa industrial debió completar, evidenciando la granularidad del reporte ESG requerido.

Tras llenar los datos de **ESRS E1**, el gerente exploró brevemente otros apartados de “Medición” (por ejemplo, indicadores sociales básicos), aunque sin completar todos los campos debido al **tiempo limitado y a la falta de ciertos datos a mano**. Finalmente,

navegó a la sección “**Generar Informe**” para ver **cómo la plataforma compilaba la información ingresada en un borrador de informe**. Previo a generar el reporte, también revisó la pestaña de “**Resultados**” que mostraba un resumen del progreso con *ESRS E1* marcado como completado y otros estándares aún pendientes confirmando que el sistema llevaba un control interno de la completitud de los datos ingresados.

Feedback cualitativo y hallazgos: En general, la reacción del usuario de la pequeña industria fue **positiva** en cuanto a la utilidad del prototipo. Manifestó que la plataforma **le permitió entender con más claridad** qué información debería recopilar su empresa para un eventual informe de sostenibilidad. Comentó, por ejemplo: *“La lista de temas y el formulario me sirvieron de guía, ahora sé qué preguntas hacernos en la empresa para cumplir con estos estándares.”* Destacó la **organización estructurada** del contenido y la interfaz relativamente intuitiva, comparándola favorablemente con la alternativa de **usar hojas de cálculo dispersas**. En particular, valoró que cada punto de divulgación estuviera identificado por su código (E1-1, E1-2...), lo cual *“facilita referenciar los requisitos exactos sin ambigüedad”*. Por otro lado, también surgieron **dificultades y sugerencias**. Si bien el usuario pudo completar los campos, señaló que **algunos términos le resultaron técnicos** o poco conocidos (por ejemplo, *“remuneración de la alta gerencia ligada a criterios climáticos”*), lo que le llevó a dudar sobre cómo responder adecuadamente. Este feedback sugiere que sería beneficioso incorporar tooltips, glosarios o explicaciones contextuales en la interfaz para cada indicador, facilitando la comprensión a usuarios de PYMEs con menor familiaridad en reportes ESG.

Desde el punto de vista **técnico**, el usuario de la PYME **no encontró fallos graves** en el funcionamiento general. La aplicación respondió fluidamente durante la mayor parte de la sesión; los tiempos de carga al navegar entre secciones fueron percibidos como breves y aceptables. Sin embargo, se identificó **un detalle de usabilidad**: al terminar de completar un estándar y regresar a la vista de “Temas Clave” o “Resultados”, el gerente no tenía claro visualmente cuáles secciones faltaban por completar, ya que el indicador de estado (completado/pendiente) no era muy llamativo. Propuso que **se resaltara con colores más visibles** (p. ej., verde y rojo) o se mostrara un porcentaje global de avance del informe para

mejorar la **retroalimentación visual** sobre el progreso. Asimismo, sugirió habilitar una opción para marcar ciertos estándares como “no aplicables” en caso de PYMEs a las que algunos requerimientos no correspondan por su tamaño o sector funcionalidad que el prototipo aún no contemplaba en esta versión. A pesar de esas observaciones, el usuario expresó **satisfacción global** con la herramienta, afirmando que *“nos ahorraría mucho tiempo y nos orientaría bastante si tuviéramos que hacer un informe de sostenibilidad el año que viene”*.

En síntesis, los **hallazgos clave** de esta validación con la empresa industrial pequeña fueron:

- **Utilidad percibida:** La plataforma guía efectivamente a usuarios sin experiencia en ESG, ayudándoles a entender **qué información recopilar y reportar**.
- **Fortalezas de la interfaz:** Estructuración clara por estándares, navegación fluida y uso de identificadores normativos precisos.
- **Áreas de mejora:** Proveer más ayuda contextual sobre términos técnicos; mejorar la visualización del progreso y permitir marcar elementos no aplicables para adaptarse a la realidad de algunas PYMEs.
- **Aspectos técnicos:** Rendimiento adecuado del prototipo; sin errores graves, aunque podría mejorarse la **experiencia de usuario** con indicadores visuales más claros y mensajes de confirmación (por ejemplo, al guardar datos con éxito).

9.3 VALIDACIÓN EN UNA STARTUP DE SOSTENIBILIDAD

Perfil de la empresa y usuarios: El segundo piloto se realizó con una **empresa emergente orientada a la sostenibilidad**, de carácter tecnológico. Se trata de una *startup* fundada hace 2 años que desarrolla soluciones de eficiencia energética para hogares, con unos 15 empleados. Aunque es una compañía de menor tamaño, su misión intrínseca es sostenible, por lo que **la medición de impacto y los criterios ESG forman parte de su**

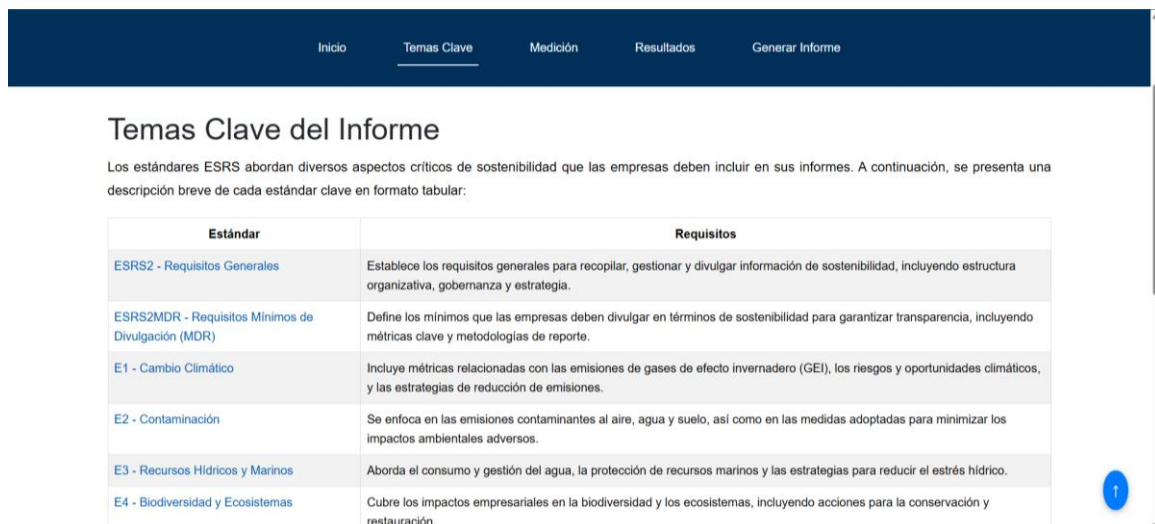
estrategia desde el inicio. La empresa busca posicionarse ante inversionistas de impacto, por lo que ya había considerado voluntariamente la elaboración de informes de sostenibilidad básicos. En la sesión de validación participaron **dos personas:** la **Responsable de Impacto y Sostenibilidad** (una de las cofundadoras, encargada de reportar métricas de impacto a inversores) y un **Analista Junior** recientemente incorporado, que apoya en la recopilación de datos ambientales y sociales. Ambos usuarios presentan un **perfil joven y tecnológicamente familiarizado** con herramientas digitales, lo que aportó una perspectiva crítica sobre la experiencia de usuario y posibles integraciones con otros sistemas.

Entorno de prueba: Esta prueba piloto se llevó a cabo de forma **remota** mediante videoconferencia. Los usuarios accedieron al prototipo a través de sus navegadores web, mientras el facilitador (investigador) observaba la sesión compartiendo pantalla y tomando notas. La duración total fue de alrededor de 1,5 horas. Previamente se envió a los participantes un breve instructivo por correo sobre cómo acceder al sistema y se les indicó el objetivo: **evaluar la plataforma simulando que preparaban el informe ESG anual de su startup.** Durante la sesión, se fomentó que los usuarios **pensaran en voz alta**, comentando sus impresiones mientras navegaban; esto permitió recopilar observaciones espontáneas sobre la usabilidad. Al final, se realizó una breve entrevista estructurada para profundizar en ciertos comentarios y consolidar la retroalimentación.

Tareas realizadas: A diferencia del caso anterior, los usuarios de la startup siguieron un flujo de trabajo **menos lineal y más exploratorio.** Iniciaron en la página de **Inicio** del prototipo, donde se muestra información general sobre la herramienta y el contexto regulatorio. Les llamó positivamente la atención que la plataforma proporcionara **insights** introductorios sobre la importancia del ESG y las nuevas normativas, lo cual “*ayuda a enmarcar por qué hacemos esto*”, según palabras de la cofundadora. Tras esta introducción, pasaron a inspeccionar el menú principal y las distintas secciones disponibles: “**Temas Clave**”, “**Medición**”, “**Resultados**” y “**Generar Informe**”. Los usuarios navegaron **de un apartado a otro para familiarizarse con la interfaz**, antes de ingresar datos. Este comportamiento evidenció su interés en comprender la **estructura**

global de la aplicación antes de centrarse en detalles. Por ejemplo, abrieron la sección “Temas Clave del Informe” para ver la lista de estándares y rápidamente cambiaron a “Medición” para revisar cómo eran los formularios de entrada. Luego volvieron a “Temas Clave” y marcaron mentalmente qué estándares serían relevantes para su startup (dado su sector tecnológico, identificaron mayor relevancia en temas de gobernanza y capital humano, además de clima).

Figura 32: Pantalla principal mostrando la navegación por secciones y la lista de “Temas Clave del Informe”



Fuente: Elaboración propia

En esta captura se aprecian el menú superior (Inicio, Temas Clave, Medición, Resultados, Generar Informe) y la tabla de estándares ESG con su descripción y estado. Esta imagen ilustró la facilidad con que los usuarios de la startup pudieron visualizar todos los temas a cubrir y navegar entre secciones para planificar su trabajo de reporte.

Tras el recorrido inicial, la Responsable de Sostenibilidad procedió a **introducir información en la sección “Medición”**, enfocándose en áreas donde ya disponían de datos internos. Por ejemplo, completó apartados sobre **políticas de diversidad e inclusión** (ESRS S – Social) y **buen gobierno corporativo** (ESRS G – Gobernanza), describiendo

las iniciativas de su empresa en estos campos. El Analista Junior, por su parte, ingresó datos ambientales sencillos, como el cálculo aproximado de las **emisiones de GEI** evitadas por el producto de eficiencia energética que comercializan (un dato que la startup ya comunicaba a sus inversores). Ambos usuarios trabajaron en paralelo en el mismo formulario, alternando el control, lo cual permitió también evaluar la **experiencia colaborativa** con un solo usuario a la vez en la interfaz (se observó que, si bien la plataforma es **monousuario** en esta versión, el equipo pudo coordinarse verbalmente para llenar distintos campos uno tras otro sin inconvenientes).

Durante la introducción de datos, los usuarios exploraron cómo la plataforma manejaba **validaciones y ayudas**: probaron ingresar valores fuera de rango o texto donde se esperaba un número, comprobando que en algunos casos el sistema ya tenía controles (no aceptó porcentajes superiores a 100% en cierto campo), mientras que en otros campos no había restricciones claras, permitiendo, por ejemplo, dejar un campo numérico vacío sin advertencia. Este comportamiento fue deliberado para **estresar el prototipo** y descubrir posibles mejoras. De hecho, encontraron un **pequeño error**: al cambiar rápidamente de estándar dentro de “Medición” sin guardar manualmente los datos, si regresaban luego al estándar anterior, la información recién escrita se había perdido. Detectaron así la ausencia de un **guardado automático o aviso de cambios no guardados**, feedback que remarcaron como importante para una próxima iteración. El equipo de la startup también verificó la sección “Resultados” tras ingresar varios datos, notando que el estado de los estándares se actualizaba a “Completado” correctamente una vez llenados todos sus campos obligatorios. Finalmente, generaron un **informe preliminar** mediante la pestaña “Generar Informe”. Dado que no habían completado absolutamente todos los estándares, la plataforma igualmente produjo un documento borrador con la información disponible. Los usuarios **descargaron el PDF del informe** generado para revisarlo en detalle.

Feedback cualitativo y hallazgos: Los participantes de la startup proporcionaron **retroalimentación abundante**, alineada con su perfil tech y su familiaridad con productos digitales. En términos generales, el prototipo fue **bien recibido por su concepto y alcance**: la cofundadora destacó que *“tener una herramienta específica nos simplifica mucho la*

vida; esto traduce requisitos complejos en pasos manejables". Ambos apreciaron la **experiencia de usuario sencilla y moderna**, señalando que la interfaz *web* les resultó limpia y comprensible sin necesidad de capacitación extensa. La **navegación entre secciones** fue identificada como uno de los puntos fuertes; según indicaron, les permitió moverse lógicamente *"de un panorama general (Temas Clave) a los detalles (Medición) y finalmente al resultado final (Informe)"*, reflejando un flujo intuitivo. En particular, valoraron la **vista de "Temas Clave"** por dar una visión clara del universo de contenidos ESG que debían abordar, lo cual les ayudó a planificar qué información necesitarían recopilar de distintos equipos (ej.: datos de RR.HH. para métricas sociales, datos financieros para gobierno corporativo, etc.).

Por otro lado, sus comentarios identificaron **oportunidades de mejora y posibles funcionalidades adicionales**. En cuanto a **usabilidad**, sugirieron incorporar una función de **autosave** o de alerta al navegar fuera de una sección con datos sin guardar, para prevenir pérdidas de información (esto derivado del pequeño incidente detectado durante la prueba). También propusieron mejorar la **edición colaborativa**, por ejemplo, permitiendo que múltiples usuarios puedan trabajar simultáneamente en diferentes secciones con cuentas separadas, ya que en su dinámica real varios miembros del equipo contribuirían al informe. Otra recomendación fue añadir algún tipo de **dashboard inicial** con indicadores resumen o un **checklist** de progreso global, para que al entrar a la plataforma se visualice inmediatamente cuánto del informe está completo. Desde el punto de vista del **contenido**, la startup encontró muy completos los campos basados en ESRS, pero comentó que quizás les gustaría poder **agregar información narrativa adicional** en el informe (una introducción de la empresa, o destacar casos de éxito en sostenibilidad) para personalizar el reporte final más allá de simplemente cumplir requisitos. Esta observación apunta a la necesidad de equilibrar entre **estructura estandarizada vs. flexibilidad** para personalización en informes de sostenibilidad.

En el aspecto **técnico y de integraciones**, dado su perfil tecnológico, los usuarios imaginaron también futuras mejoras: por ejemplo, la posibilidad de **importar automáticamente ciertos datos** desde fuentes existentes (como hojas de cálculo de

emisiones o sistemas de recursos humanos) para evitar carga manual, o la integración con APIs que actualicen indicadores en tiempo real. Reconocieron que estas características podrían estar fuera del alcance del prototipo actual, pero las mencionaron como dirección de desarrollo. Asimismo, evaluaron el **rendimiento**: en general la aplicación respondió **ágilmente**, incluso al alternar secciones y generar el PDF, que se descargó en pocos segundos. No experimentaron caídas ni errores críticos, más allá del asunto del guardado ya mencionado.

En síntesis, la validación con la startup evidenció los siguientes puntos relevantes:

- **Confirmación de usabilidad:** La plataforma resultó intuitiva y agradable de usar para un público joven y habituado a herramientas digitales, validando que el **diseño UX** es acertado en términos generales.
- **Valor percibido:** Se destacó el valor de convertir **requisitos complejos (ESRS)** en un proceso guiado. Los usuarios ven la herramienta como un facilitador para que pequeñas empresas innovadoras puedan cumplir estándares de reporting sin tener expertos dedicados.
- **Incidencias y mejoras:** Se descubrió un fallo menor (pérdida de datos no guardados al cambiar de sección), llevando a recomendar funcionalidad de guardado automático/alertas. También se sugirió habilitar trabajo colaborativo multiusuario y mayor personalización del informe final, para acercar el prototipo a un producto listo para entornos profesionales reales.
- **Extensibilidad:** Los participantes vislumbraron potencial de crecimiento (integraciones con otras plataformas, automatización de importación de datos), indicando que existe interés en seguir utilizando una herramienta así si evoluciona con funcionalidades avanzadas.

9.4 EMPRESA MANUFACTURERA MEDIANA

Perfil de la empresa y usuario: El tercer caso de validación se centró en Industrias Mercisa S.A., una mediana empresa manufacturera española del sector automotriz con unos 150 empleados. Mercisa se dedica a la producción de componentes metálicos para vehículos y cuenta con certificaciones de calidad (ISO 9001) e interés reciente en sostenibilidad. El evaluador fue el Jefe de Calidad y Medioambiente (perfil técnico, ingeniero industrial) de la compañía, responsable también de los temas de RSC. Este usuario posee experiencia en gestión de calidad y procesos industriales, pero tiene conocimientos limitados en estándares ESG completos. Su perfil representa bien a muchas pymes manufactureras que requieren apoyo para abordar los nuevos requisitos de reporte.

Entorno de prueba: La validación se realizó presencialmente en las oficinas centrales de Mercisa. Se habilitó un equipo portátil con el prototipo web cargado en un navegador reciente. Antes de comenzar, se brindó al evaluador una breve introducción sobre el objetivo de la plataforma y el contexto normativo (CSRD/ESRS). Posteriormente, el usuario interactuó libremente con la demo como si preparara un informe real, siguiendo las secciones propuestas por la aplicación. El investigador observó discretamente, interviniendo sólo para resolver dudas puntuales. En todo momento se destacó que esta sesión correspondía a una fase piloto temprana del proyecto, diseñada para recoger retroalimentación que guíe el desarrollo completo de la herramienta.

Tareas realizadas: El flujo de trabajo fue lineal y guiado. El evaluador inició en la página de inicio consultando la descripción general de la plataforma, para luego acceder a “**Temas Clave del Informe**”. Allí encontró listados los diferentes estándares ESG aplicables (ESRS generales, medioambientales, sociales y de gobernanza), cada uno con su descripción y su estado de cumplimiento. Observó que la navegación entre secciones se comportaba como en una SPA: la función JavaScript `showSection(id)` permitía alternar las vistas dinámicamente sin recargar la página. Esta estructura le ofreció un panorama general útil para planificar el reporte. A continuación, el evaluador ingresó a la sección “**Medición**” y seleccionó los estándares más relevantes para su empresa (por ejemplo,

ESRS El cambio climático, dada la disponibilidad de datos de consumo eléctrico y emisiones, y otros indicadores ambientales de interés). En cada formulario de medición comenzó a introducir información: los campos de respuesta eran editables directamente en la tabla (se implementó el atributo HTML5 `contenteditable="true"` en las celdas de respuesta), de modo que podía hacer clic en cada celda y teclear datos o texto libre de forma intuitiva. Conforme completaba los campos, notó las **barras de progreso** que aparecían en la parte superior de cada formulario, indicando el porcentaje de avance en tiempo real. Por ejemplo, al introducir el consumo eléctrico anual, el indicador mostró inmediatamente el avance correspondiente. El usuario completó tanto campos numéricos (consumos, valores de emisiones) como cualitativos (descripciones de iniciativas ambientales), alternando entre secciones para revisar el flujo general y entender el mecanismo de guardado (que en esta demo se hacía manualmente al cambiar de formulario).

Feedback cualitativo y hallazgos: En su balance final, el responsable de calidad valoró positivamente la utilidad de la herramienta en un entorno manufacturero. Comentó que la plataforma ofrecía una **guía estructurada** que facilita saber qué información recopilar para un informe de sostenibilidad: *“Tener una guía paso a paso nos ayuda a centrarnos en lo más importante sin perdernos”*, indicó. Destacó que automatizaría y agilizaba el proceso frente a métodos basados en hojas de cálculo dispersas, ya que consolida las preguntas ESG en un solo lugar. Asimismo, apreció la fluidez de la interfaz y el diseño claro: la posibilidad de editar directamente en tabla resultó natural, y el menú superior permitió moverse lógicamente de un panorama general (Temas Clave) a los detalles (Medición).

Como sugerencias de mejora, el evaluador señaló varios puntos. Propuso resaltar de forma más visible las secciones incompletas (usando colores llamativos o mostrando un **porcentaje global de avance** del informe) para saber rápidamente qué falta. Además, consideró útil permitir marcar ciertos estándares como **“no aplicables”** si la empresa no debe reportarlos, así como incorporar ayudas contextuales o glosarios emergentes para términos muy técnicos. También sugirió implementar un autoguardado de datos al cambiar de sección para evitar pérdidas accidentales. Dado que en la práctica participaría un equipo, mencionó que en iteraciones futuras sería conveniente habilitar múltiples usuarios con roles

diferenciados (colaboración simultánea). En general, el usuario confirmó que esta validación formaba parte de un piloto previo al desarrollo completo de la plataforma, por lo que sus comentarios serán de utilidad para las siguientes iteraciones.

En conclusión, el evaluador de Mercisa coincidió en que el prototipo ya aporta beneficios concretos para el reporte ESG en manufactura (eficiencia, estructura normativa y consistencia), y sus observaciones se han anotado para mejorar la versión final de la herramienta. Integrar estas mejoras (ayudas en contexto, indicadores de progreso más claros, opciones de personalización) contribuirá a que la aplicación sea plenamente útil y adaptable a entornos industriales reales.

Referencias de funcionalidades implementadas: La validación resaltó aspectos clave de la implementación técnica: la navegación SPA de la aplicación (sin recargas de página al cambiar de sección), los campos editables con `contenteditable` en los formularios, y las barras de progreso independientes en cada formulario para mostrar el avance. Estos elementos reales del prototipo influyeron positivamente en la experiencia del usuario, y sus comentarios permitirán afinar la solución antes de su despliegue final.

9.5 SÍNTESIS DE LA VALIDACIÓN Y RECOMENDACIONES FINALES

Los tres casos de uso simulados en este capítulo proporcionan una **validación integral del prototipo** abarcando distintos contextos empresariales. En conjunto, las pruebas piloto confirmaron la **aplicabilidad** de la plataforma de reportes ESG para **pequeñas y medianas empresas**, evidenciando que tanto organizaciones con poca experiencia en sostenibilidad como startups especializadas y compañías medianas con práctica en RSC **pueden beneficiarse** de su uso. Cada perfil de usuario encontró valor en la herramienta: ya sea como **guía educativa** para iniciarse en el reporting (caso de la PYME industrial), como **apoyo estructural y tecnológico** para potenciar iniciativas existentes (caso de la startup), o como **solución de eficiencia y control de calidad** en un proceso formal de reporte (caso de la empresa mediana).

Asimismo, las validaciones permitieron identificar **puntos fuertes transversales** del prototipo. Entre ellos destacan la **estructura clara por secciones** (facilitando navegar del panorama general al detalle), la **alineación con estándares reconocidos (ESRS)** asegurando relevancia regulatoria, y la **generación automática de informes** que ahorra tiempo y provee un resultado consistente. Estos atributos fueron valorados repetidamente por los usuarios piloto. Del mismo modo, se observaron **temáticas comunes de mejora**. Varias sugerencias convergieron en la necesidad de aumentar la **usabilidad y flexibilidad**: por ejemplo, ofrecer ayudas contextuales, funcionalidades de guardado automático, capacidades multiusuario y opciones de personalización del informe. También se identificó la importancia de pulir ciertos **aspectos técnicos** propios de un prototipo, como la validación de entradas de datos, la gestión de errores menores (formato de tablas) y la optimización para evitar pérdidas de información.

En términos prácticos, a partir de esta simulación de validación se recomiendan las siguientes **acciones concretas** para futuras iteraciones del sistema:

- **Incorporar mejoras de UX/UI:** incluir descripciones o glosarios in-app para términos complejos, destacar visualmente el progreso de llenado y notificar al usuario sobre datos sin guardar o secciones incompletas.
- **Agregar funcionalidad colaborativa:** habilitar cuentas de usuario múltiples con roles/permisos, y considerar bloqueo de secciones en edición para permitir que equipos trabajen simultáneamente sin interferir.
- **Permitir edición post-generación:** ofrecer el informe final en un formato que admita ajustes manuales (o un módulo de edición previa a la exportación) para que las empresas puedan personalizar la presentación final sin perder el formato base.
- **Refinar el motor de reporte:** solucionar los detalles de paginación y formato identificados, e incorporar plantillas flexibles que se adapten a distintos volúmenes de contenido.

- **Mantener actualización de contenido:** establecer un procedimiento para actualizar los estándares ESG en la plataforma (nuevos ESRS o modificaciones) de forma ágil, garantizando que la herramienta siga siendo útil conforme evolucionen las normativas.

Expresado de forma concisa, la validación demuestra que el prototipo desarrollado es **viable y valioso** para su público objetivo (PYMEs españolas) y cumple con su objetivo de facilitar la elaboración de informes de sostenibilidad. Las interacciones simuladas con usuarios reales confirmaron los **beneficios clave** de la herramienta en términos de eficiencia, guía normativa y estandarización, al mismo tiempo que proporcionaron **insumos valiosos para perfeccionar** el sistema. Con las mejoras sugeridas muchas de las cuales son abordables gracias al diseño modular y escalable descrito en capítulos anteriores, el prototipo podría evolucionar hacia una plataforma robusta de uso extendido, contribuyendo a que más pequeñas y medianas empresas puedan afrontar los retos del **reporting** ESG de forma eficaz y alineada con las mejores prácticas internacionales. Los resultados de estas pruebas piloto, por lo tanto, no solo validan la propuesta inicial, sino que orientan el **trabajo futuro** para llevar esta herramienta del estado de prototipo a un **producto plenamente implementable** en entornos empresariales reales.

Capítulo 10. DESPLIEGUE TÉCNICO SENCILLO

En capítulos previos de este trabajo se detalló el desarrollo de una aplicación web de página única (SPA) realizada con HTML, CSS y JavaScript, diseñada para ayudar a las pymes en la preparación de informes de sostenibilidad conforme a la directiva europea CSRD. Dado que la aplicación carece de backend (toda la lógica reside en el navegador del cliente), su despliegue técnico puede abordarse de forma particularmente simple. En esta sección se presenta una hoja de ruta de despliegue técnico sencillo, exponiendo los pasos concretos para publicar la herramienta de manera realista en el mundo real. El objetivo es lograr que la aplicación sea fácilmente accesible para los usuarios finales (pymes interesadas en la sostenibilidad), minimizando la complejidad técnica y el coste inicial. Se abordará primero un despliegue inicial utilizando servicios gratuitos, seguido de consideraciones sobre futuras mejoras (uso de dominio propio, posible incorporación de un backend, escalabilidad y mantenimiento) y recomendaciones para un despliegue profesional eventual. Esta hoja de ruta se describe con rigor técnico, pero en un tono accesible, asumiendo conocimientos básicos en informática por parte del lector.

10.1 HOJA DE RUTA DE DESPLIEGUE INICIAL

A continuación, se describe paso a paso el plan de despliegue inicial de la aplicación, haciendo uso de GitHub y GitHub Pages para lograr una publicación rápida y gratuita de la versión demo.

1. **Registro del repositorio y control de versiones:** El primer paso consiste en crear un **repositorio** para el proyecto en la plataforma GitHub. Un repositorio es un espacio de almacenamiento de código fuente en la nube que permite llevar un **control de versiones** del software, es decir, registrar los cambios a lo largo del tiempo y coordinar contribuciones. El autor creará un repositorio (de preferencia público para facilitar el acceso) bajo su cuenta de GitHub, subiendo todos los

archivos de la aplicación al mismo. Se recomienda organizar el código con una estructura clara de carpetas, de modo que el proyecto sea fácil de mantener y escalar. Por ejemplo, una estructura adecuada podría ser la siguiente: un archivo principal `index.html` (que cargará la SPA), una carpeta `styles/` con las hojas de estilo CSS, una carpeta `scripts/` con los archivos JavaScript de la lógica de la aplicación, y una carpeta `assets/` para imágenes, fuentes u otros recursos estáticos. Esta organización separa correctamente el HTML, los estilos y los recursos multimedia, lo que **facilita la navegación y el mantenimiento del código**. Una vez estructurado el repositorio, GitHub se encargará de almacenar cada versión mediante **commits**, permitiendo deshacer cambios si es necesario y proporcionando transparencia sobre la evolución del código. Este enfoque de control de versiones no solo ayuda en el desarrollo individual sino que, a futuro, facilitaría la colaboración con otros desarrolladores si el proyecto creciera.

2. **Publicación en GitHub Pages:** Con el código fuente en GitHub, el siguiente paso es aprovechar **GitHub Pages**, un servicio gratuito de GitHub para alojar sitios web estáticos directamente desde el repositorio. GitHub Pages es especialmente adecuado para este proyecto porque **es gratuito y fácil de configurar**, lo que lo hace muy atractivo para prototipos o proyectos pequeños. Al ser un servicio de alojamiento estático, **no requiere configurar servidores ni bases de datos**, basta con subir los archivos HTML, CSS y JS al repositorio para tener el sitio funcionando. En la práctica, el autor activará GitHub Pages en la configuración del repositorio: esto puede hacerse utilizando la rama principal (`main`) o creando una rama dedicada (por convención llamada `gh-pages`) donde residan los archivos estáticos. La forma sencilla es habilitar la publicación desde la rama principal (o una carpeta designada como `/docs`) mediante la pestaña de **Settings > Pages** en GitHub. Una vez habilitado, GitHub generará una URL pública para la aplicación, típicamente del estilo <https://usuario.github.io/nombre-del-repositorio/>. En esta dirección web quedará accesible la herramienta para cualquier usuario con conexión a Internet. Este enfoque permite **poner en línea una demo pública sin**

costo y de forma inmediata, ideal para propósitos de validación inicial con pymes. Se debe tener en cuenta que GitHub Pages **solo admite contenido estático**, por lo que esta solución es viable gracias a que nuestra aplicación es una SPA puramente del lado cliente (sin componentes dinámicos de servidor). Si más adelante se necesitaran funcionalidades dinámicas (por ejemplo, almacenamiento en base de datos o autenticación de usuarios), sería preciso migrar a otra infraestructura, como se discutirá en secciones posteriores. Para la demo actual, GitHub Pages ofrece suficiente capacidad; solo habría que vigilar sus límites de uso en caso de un tráfico inusualmente alto, ya que existen restricciones de ancho de banda y espacio en este servicio (pensado para proyectos de tamaño moderado).

3. **Landing page explicativa vinculada a la herramienta:** Como parte del despliegue inicial, se preparará una **landing page** o página de aterrizaje que sirva de introducción a la aplicación. Esta página de bienvenida, que será la primera que vean los visitantes en la URL publicada, explicará en términos sencillos el propósito y funcionalidad de la herramienta. Incluirá una breve descripción del problema que resuelve (como la dificultad que enfrentan las pymes para generar informes de sostenibilidad conforme a la CSRD) y cómo la aplicación les ayuda a solucionarlo. También podría mostrar las principales características, capturas de pantalla ilustrativas y guías de uso básicas. La landing page estará **vinculada al índice de la SPA**; es decir, contendrá un botón o enlace prominente de "Entrar a la aplicación" o "Probar la herramienta" que llevará al usuario a la interfaz principal (el formulario o panel donde la pyme introduce datos para generar su informe). Dado que en una SPA típicamente toda la aplicación reside en una sola página `index.html`, hay dos enfoques posibles: (a) integrar la sección explicativa en la propia `index.html` mostrando primero la información introductoria y luego permitiendo iniciar la interacción, o (b) tener una página separada (por ejemplo `landing.html`) con la explicación y un enlace que abra la aplicación en `index.html`. En este caso, se optará por la primera opción por simplicidad: el archivo `index.html` actuará como landing page inicial y también contendrá la aplicación, estructurando su contenido en

secciones. Al entrar, el usuario verá la explicación y luego podrá desplegar el formulario interactivo de la herramienta sin cambiar de página. Esta estrategia aprovecha la naturaleza de SPA, evitando redirecciones y manteniendo todo en un mismo lugar. La página de aterrizaje deberá ser **clara y atractiva**, utilizando los elementos de diseño adecuados (títulos, subtítulos, quizás el logo de la aplicación si lo hubiera, y estilos coherentes con la identidad visual del proyecto) para causar una buena impresión y fomentar la confianza del usuario. Además, incluirá información de contacto o enlace a más documentación (por ejemplo, un enlace al repositorio de GitHub para los interesados en el código fuente o a un manual de usuario si se dispone de él). Tras diseñar esta landing page, se subirá junto con el resto de archivos al repositorio, de modo que quede publicada automáticamente en GitHub Pages. En suma, la landing page funciona como una **porta de entrada explicativa** que contextualiza la herramienta antes de que la pyme comience a usarla, asegurando que incluso lectores no técnicos comprendan de qué se trata y cómo empezar.

Con los pasos anteriores, el despliegue inicial quedaría completado: la aplicación estaría en línea en la URL de GitHub Pages, con su código fuente versionado en GitHub y una página inicial informativa. En este punto, el autor puede compartir la URL con interesados (responsables de sostenibilidad en pymes, tutores académicos...) para que prueben la demo. El mantenimiento en esta fase es muy sencillo: cualquier mejora o corrección en el código simplemente requiere hacer un nuevo commit y push al repositorio; GitHub Pages actualizará automáticamente el sitio con la última versión. No hay servidores que monitorear ni bases de datos que administrar, lo que reduce significativamente la carga de mantenimiento inicial. A continuación, se discuten posibles evoluciones futuras del despliegue, en caso de que el proyecto crezca en alcance o requiera funcionalidades adicionales.

10.2 POSIBLE MIGRACIÓN A DOMINIO PROPIO Y SERVIDOR

Si el proyecto evoluciona favorablemente (por ejemplo, aumentando su base de usuarios o pasando de fase demo a un servicio más formal), sería conveniente considerar una migración a un dominio propio y, de ser necesario, a un servidor dedicado. **Inicialmente, para mejorar la imagen y accesibilidad, se puede adquirir un nombre de dominio personalizado** (por ejemplo, un .com o .es representativo del producto) y apuntarlo al sitio de GitHub Pages. GitHub Pages permite asociar dominios personalizados de manera sencilla, lo que significa que podríamos tener una dirección web profesional (como www.sostenidata.com) sin abandonar la infraestructura gratuita. Esta opción brinda una presencia en línea más seria sin incurrir aún en costes de hosting, ya que GitHub seguiría sirviendo el contenido estático. La configuración implica comprar el dominio en un registrador, luego configurar un registro DNS (tipo CNAME o A) que dirija al dominio de GitHub Pages. Tras unos ajustes en la configuración del repositorio (añadiendo el dominio personalizado), el sitio quedaría accesible mediante la nueva URL.

Pero **si el proyecto requiere capacidades más allá de lo estático**, como interacción con una base de datos, autenticación de usuarios, envío de correos, generación de informes en el lado servidor, etc., entonces GitHub Pages resultaría insuficiente y habría que **migrar a un entorno con servidor**. En esta etapa, el autor evaluaría opciones de hosting web convencional o servicios en la nube. Una alternativa sencilla es contratar un **servidor de alojamiento compartido** que soporte tecnologías web (muchos planes de hosting económicos soportan directamente sitios estáticos e incluso lenguajes como PHP). Otra opción más flexible es desplegar la aplicación en un **servidor virtual privado (VPS)** o en servicios cloud tipo Plataforma-como-Servicio (PaaS). Por ejemplo, servicios en la nube como Heroku, Netlify, Vercel, Firebase Hosting u otros podrían facilitar el despliegue con mayor funcionalidad (algunos ofrecen alojamiento estático con funciones adicionales, y otros permiten desplegar aplicaciones Node.js o contenedores con backend). La decisión dependerá de las necesidades específicas: si solo se desea un dominio propio y mantener la aplicación estática, GitHub Pages con dominio customizado sería suficiente; pero si se

busca ampliar las funcionalidades técnicas, migrar a un servidor con capacidades dinámicas será inevitable.

La migración a un servidor propio implicaría **transferir los archivos de la aplicación a un servidor web** (por ejemplo, Apache o Nginx) configurado en el hosting escogido. Dado que actualmente la aplicación es estática, inicialmente podría seguir funcionando tal cual en el nuevo servidor (sirviendo el `index.html` y archivos asociados). Sin embargo, disponer de un servidor abre la puerta a implementar mejoras del lado del servidor (discutidas en la siguiente sección). Cabe destacar que muchos hostings permiten subir el sitio vía FTP o sistemas de integración continua, pero si el repositorio GitHub está bien mantenido, se puede conectar ese flujo de trabajo: por ejemplo, usar acciones de GitHub (GitHub Actions) para desplegar automáticamente en el nuevo servidor tras cada commit. En cualquier caso, durante esta transición habría que planificar adecuadamente la **configuración del entorno** (sistemas operativos, seguridad básica del servidor, certificados SSL para el nuevo dominio, etc.) ya que son aspectos que en GitHub Pages estaban abstraídos. Para mantener las buenas prácticas, se habilitaría HTTPS en el dominio propio (posiblemente usando certificados gratuitos como Let's Encrypt) para asegurar la comunicación cifrada con los usuarios.

En forma sintética, el paso a un dominio y servidor propios proporciona mayor control y profesionalidad al despliegue, pero también conlleva encargarse de la infraestructura. Por ello, es recomendable abordar esta migración solo cuando las necesidades lo justifiquen (como una demanda de usuarios que requiera branding y funcionalidades avanzadas). Mientras tanto, la solución inicial (GitHub Pages) es suficiente y eficiente para la fase de demostración.

10.3 EVOLUCIÓN HACIA UN BACKEND (NODE.JS, PHP, BASE DE DATOS)

Actualmente, la herramienta funciona completamente en el navegador del usuario, lo que significa que **no hay una base de datos ni lógica de negocio en servidor**. Esta arquitectura

sin backend simplifica el despliegue, pero impone limitaciones: por ejemplo, los datos que introduce la pyme no se almacenan en ningún servidor (solo existen temporalmente en el navegador mientras la sesión está abierta), y funcionalidades como autenticación de usuarios, almacenamiento persistente de informes o generación de contenidos bajo demanda del lado del servidor no son posibles. Si en el futuro se quisiera añadir alguna de estas capacidades, sería necesario desarrollar un **componente de backend** e integrarlo con la aplicación.

Existen varias rutas tecnológicas para implementar un backend. Una opción moderna y coherente con el stack actual es usar **Node.js**, que permitiría programar el servidor en JavaScript (el mismo lenguaje ya empleado en el frontend). Node.js es un entorno de ejecución de JavaScript del lado del servidor, ampliamente utilizado para construir servidores web escalables y APIs. Con Node.js se puede crear fácilmente un servidor que atienda peticiones HTTP, manejar usuarios (login, registro), exponer servicios REST para intercambiar datos y conectar con bases de datos. De hecho, Node.js se destaca por posibilitar entornos altamente escalables en aplicaciones web y móviles, incluyendo la creación de APIs RESTful y conexión a bases de datos SQL o NoSQL. Gracias a su naturaleza asíncrona y a la gran cantidad de librerías disponibles (frameworks como Express.js), Node.js facilitaría añadir funcionalidades como: guardar en un servidor los datos de sostenibilidad ingresados por cada pyme, generar informes PDF en el servidor, o implementar un módulo de autenticación para que cada pyme tenga su cuenta y sus informes guardados. Si se tomara este camino, el despliegue cambiaría: habría que alojar la parte de servidor de Node.js en algún servicio (por ejemplo, en Heroku o en un VPS con Node instalado) y probablemente modificar la aplicación cliente para comunicarse con ese servidor (vía llamadas a API REST).

Otra posibilidad es utilizar **PHP**, un lenguaje de scripting del lado del servidor muy consolidado en el desarrollo web tradicional. PHP ha sido durante años uno de los lenguajes backend más usados, en parte por su facilidad de uso y en parte porque forma parte del stack LAMP (Linux, Apache, MySQL, PHP) presente en la mayoría de servidores de alojamiento web. Una ventaja práctica de PHP es su **amplia disponibilidad en servicios**

de hosting: la mayoría de planes de alojamiento compartido soportan PHP de serie, lo que hace que desplegar una aplicación con PHP y una base de datos MySQL sea relativamente sencillo y de bajo coste. Si se optase por PHP, se podría, por ejemplo, crear un pequeño conjunto de scripts que reciban los datos del formulario de sostenibilidad (enviados mediante peticiones HTTP desde la SPA) y los guarden en una base de datos en el servidor, o generen un documento descargable. Frameworks PHP como Laravel o Symfony podrían ayudar a estructurar una posible aplicación de servidor más grande, aunque para funcionalidades puntuales no necesariamente se requeriría un framework completo. En cualquier caso, la elección de PHP podría ser adecuada si se busca rapidez con recursos limitados, dado que con unos pocos archivos PHP y una base de datos MySQL se puede montar un backend básico funcional. La contrapartida es que el desarrollo sería en un lenguaje distinto (PHP en el servidor, JavaScript en el cliente), pero esto es común en muchos proyectos web y hay abundante documentación al respecto.

En cuanto a la **base de datos**, independientemente de elegir Node.js o PHP (u otra tecnología) para el backend, será necesaria para persistir información. Para una aplicación enfocada a informes de sostenibilidad, una base de datos relacional clásica (por ejemplo, **MySQL** o **PostgreSQL**) podría servir para almacenar datos de empresas, indicadores de sostenibilidad, plantillas de informes, etc. Estas bases de datos son ampliamente soportadas tanto por Node (mediante bibliotecas npm) como por PHP (vía extensiones mysqli/PDO) y son fáciles de gestionar en la mayoría de hostings. Alternativamente, si se prevé almacenar datos de forma muy flexible (documentos JSON), se podría considerar una base NoSQL como **MongoDB**; sin embargo, para comenzar, una base SQL tradicional resulta suficiente y familiar para muchas pymes (en caso de que quieran exportar datos a Excel). Es importante planificar la estructura de la base de datos con miras a la futura **escalabilidad**: asegurar integridad de datos, índices para consultas eficientes y copias de seguridad periódicas. Un backend bien diseñado también debería incluir validación de datos en el servidor (para no confiar únicamente en las verificaciones del lado cliente) y medidas de seguridad (por ejemplo, sanitizar entradas para evitar inyección SQL en caso de usar SQL, o proteger endpoints con autenticación si corresponde).

Cabe mencionar que existen otras alternativas híbridas o de “backend as a service” que podrían evaluarse, como **Firebase** (de Google) que ofrece una base de datos en tiempo real y servicios de autenticación listos para usar con frontends JavaScript, o utilizar **Google Sheets/Drive** a través de APIs para almacenar datos de forma muy sencilla si el volumen es bajo. No obstante, estas opciones se apartan de un despliegue tradicional y podrían no ser adecuadas si la intención es eventualmente tener un producto robusto.

De modo sintético y claro, añadir un backend transformaría la aplicación en un sistema **full-stack**, aumentando su complejidad, pero también sus capacidades. La decisión entre Node.js o PHP (u otros lenguajes como Python/Flask, Java/Spring) dependerá de los recursos y conocimientos disponibles, así como de los requisitos específicos. Node.js ofrece una continuidad en JavaScript y alta escalabilidad, mientras PHP proporciona facilidad de despliegue en entornos comunes y rapidez para prototipos. En ambos casos, habría que planificar cuidadosamente el despliegue de este backend para integrarlo con la parte frontend ya existente.

10.4 ESCALABILIDAD Y MANTENIMIENTO

Un aspecto fundamental a considerar en cualquier despliegue es cómo escalar el sistema y cuánto esfuerzo de mantenimiento requerirá a largo plazo. La solución inicial propuesta (GitHub Pages con aplicación estática) destaca por su **simplicidad de mantenimiento**: al no haber servidor propio corriendo ni base de datos, prácticamente no hay elementos que administrar en producción. GitHub Pages se encarga de servir los archivos y el desarrollador solo necesita actualizar el contenido cuando haya mejoras. No hay que preocuparse de parches de seguridad de servidor, monitoreo de uptime, ni balanceo de carga en esta etapa. Esto supone un **mantenimiento mínimo**, ideal para un proyecto incipiente o un demo. Además, en términos de **escalabilidad inicial**, un sitio estático puede atender a muchos usuarios concurrentes, ya que las páginas estáticas se pueden cachear y distribuir eficientemente (GitHub Pages de hecho emplea una red CDN global para entregar el contenido). Por lo tanto, si un número creciente de pymes comenzara a usar la

herramienta simultáneamente, la infraestructura estática podría soportarlo razonablemente sin cambios inmediatos.

No obstante, existen **límites de escalabilidad en la solución estática gratuita**. GitHub Pages impone límites de ancho de banda y almacenamiento; si el tráfico se volviera muy alto (por ejemplo, decenas de miles de usuarios mensuales) o si la aplicación incluyera archivos muy pesados, podría alcanzarse esos toques, obligando a migrar a otro servicio de hosting pago. Asimismo, cualquier funcionalidad interactiva que exceda las capacidades del cliente (como comentado, autenticación, bases de datos) no escalará en GitHub Pages, porque simplemente no está soportada. Entonces, la **escalabilidad funcional** también es limitada: para escalar en cuanto a características ofrecidas, hay que pasar a una arquitectura con backend.

Suponiendo que en fases posteriores se implemente un backend en servidor, el **mantenimiento** se volverá más exigente. Habrá que encargarse de actualizar el sistema operativo del servidor, aplicar parches de seguridad a la plataforma (ya sea Node.js, PHP u otra) y a la base de datos, monitorear el uso de CPU, memoria y almacenamiento, así como implementar copias de seguridad periódicas de la base de datos. También será necesario vigilar el rendimiento: a medida que crezca la base de datos o la cantidad de peticiones concurrentes, se podría requerir optimizar consultas, agregar índices o incluso escalar verticalmente (a un servidor más potente) u horizontalmente (varios servidores detrás de un balanceador de carga). Afortunadamente, si se usa Node.js, este está diseñado para manejar muchas conexiones concurrentes de forma eficiente gracias a su modelo asíncrono; y en el caso de PHP, se puede aprovechar la tradicional pila LAMP que, bien configurada (como con caches como OPcache), puede soportar también una carga significativa. En cualquier caso, **planificar la escalabilidad desde el diseño** es importante: por ejemplo, separar componentes (frontend y backend, base de datos) de modo que se puedan escalar independientemente, y usar servicios gestionados cuando sea posible (una base de datos en la nube administrada por un proveedor podría reducir la carga de mantenimiento).

Un punto a destacar es que la escalabilidad no solo se refiere a la capacidad técnica, sino también a la facilidad de incorporar nuevos desarrolladores o nuevos requisitos en el proyecto. Mantener un repositorio organizado (como se describió en la estructura de carpetas) y documentado contribuirá a que el proyecto pueda crecer sin convertirse en inmanejable. También conviene implementar gradualmente **pruebas automáticas** a medida que se agregan componentes críticos (como tests unitarios para funciones clave de JavaScript, o tests de API para el backend) para asegurar que futuras modificaciones no rompan funcionalidades existentes, lo cual es parte de un mantenimiento saludable del software.

En líneas generales, la solución inicial ofrece escalabilidad suficiente para un alcance pequeño-moderado con un esfuerzo de mantenimiento casi nulo, mientras que las expansiones futuras hacia un backend requerirán mayor planificación para garantizar escalabilidad técnica (soportar más usuarios y datos) y asumirán una responsabilidad de mantenimiento continua. El presente plan de despliegue sencillo sienta una base sólida, pero es importante estar preparado para evolucionar la infraestructura cuando la demanda o las funcionalidades lo exijan.

10.5 RECOMENDACIONES PARA UN DESPLIEGUE PROFESIONAL FUTURO

Pensando en el largo plazo y en un entorno más profesional, se sugieren a continuación algunas recomendaciones para llevar el despliegue de la aplicación al siguiente nivel:

- **Adquisición de dominio y branding:** Como ya se indicó, registrar un dominio propio es crucial para dar credibilidad a la aplicación de cara a clientes empresariales. En un despliegue profesional, se debe asegurar que el dominio esté acompañado de un certificado SSL válido (HTTPS) para garantizar la seguridad en las comunicaciones. El sitio (ya sea estático o dinámico) debería incorporar la imagen corporativa adecuada (logotipos, colores institucionales de la pyme desarrolladora) para reforzar la confianza de las pymes usuarias.

- **Infraestructura de hosting robusta:** Evaluar proveedores de **cloud hosting** que ofrezcan alta disponibilidad y escalabilidad automática. Por ejemplo, desplegar la aplicación en servicios como AWS (Amazon Web Services), Azure o Google Cloud, donde se puede comenzar aprovechando capas gratuitas o de bajo costo y luego escalar según la carga. En caso de tener backend Node.js, servicios como **Heroku**, **Railway** o **Fly.io** (PaaS amigables) podrían simplificar el despliegue, mientras que para infra más compleja contenedores Docker y Kubernetes en la nube darían máxima flexibilidad. Un despliegue profesional también contemplaría entornos separados para desarrollo, pruebas y producción, evitando que cambios no probados afecten al entorno en producción.
- **Integración continua y despliegue continuo (CI/CD):** Implementar un pipeline de CI/CD que automatice las pruebas y el despliegue. GitHub Actions u otras herramientas (Jenkins, GitLab CI) pueden configurarse para que cada vez que se haga push de cambios al repositorio, se ejecuten pruebas automatizadas y, si todo es correcto, se publique la nueva versión de la aplicación. Esto reduce errores humanos en el proceso de despliegue y acelera la entrega de mejoras a los usuarios. En un contexto profesional, la integración continua es prácticamente un estándar para garantizar calidad y rapidez.
- **Monitoreo y analytics:** Una vez desplegada la aplicación públicamente, es recomendable integrar herramientas de **monitorización** y **analítica**. Por un lado, monitorización técnica mediante servicios como NewRelic, UptimeRobot o el propio monitoreo de la plataforma cloud elegida, para recibir alertas de caídas del sistema, altos tiempos de respuesta u otros problemas. Por otro lado, analítica de uso (Google Analytics u otras soluciones enfocadas a productos SaaS) para comprender cómo las pymes están utilizando la herramienta, qué secciones usan más, dónde podrían estar teniendo dificultades, etc. Estos datos ayudarán a mejorar tanto la infraestructura (detectando cuellos de botella) como la usabilidad del software.

- **Escalabilidad planificada y balanceo:** Si se llega a un punto de alta carga, emplear técnicas de escalabilidad horizontal, como el uso de **balanceadores de carga** (load balancers) para distribuir el tráfico entre múltiples instancias de la aplicación. Por ejemplo, en un backend Node.js se podrían correr varias instancias en paralelo (cluster de Node o pods de Kubernetes) detrás de un balanceador. En el caso de bases de datos, considerar la replicación o el uso de servicios gestionados que escalen automáticamente. También, utilizar cachés (Redis, Memcached) para aligerar la carga de operaciones repetitivas si fuese pertinente.
- **Seguridad y cumplimiento:** Un despliegue profesional debe atender seriamente la **seguridad**. Esto incluye mantener todos los componentes actualizados a nivel de parches, emplear firewalls o reglas de seguridad en servidores, y realizar pruebas de penetración o auditorías de seguridad periódicas. Dado que la aplicación trata con datos de sostenibilidad corporativos, es posible que algunos de esos datos sean sensibles; habría que asegurarse de cumplir con normativas de protección de datos (GDPR, si aplica) en caso de almacenar información identificable. Además, se debe prever un plan de recuperación ante fallos: backups regulares de bases de datos y posibilidad de restauración rápida en caso de caída (disaster recovery).
- **Soporte y mantenimiento continuo:** Finalmente, a nivel organizativo, se recomendaría definir claramente **responsables de mantenimiento** de la aplicación. En un entorno empresarial, esto podría significar tener un equipo o persona asignada para responder a incidencias, actualizar contenidos (por ejemplo, si la normativa CSRD se actualiza, la herramienta debería reflejar esos cambios) y atender dudas de los usuarios. Un buen soporte mejora la confianza de las pymes usuarias y ayuda a detectar mejoras. En paralelo, conviene mantener una documentación técnica actualizada del sistema, para que nuevos desarrolladores puedan incorporarse al proyecto y entender rápidamente la arquitectura y procedimientos de despliegue.

Siguiendo estas recomendaciones, la aplicación podrá transicionar de un prototipo funcional a un producto estable en producción, preparado para atender a un mayor número de pymes de manera confiable.

En esta sección se ha presentado un plan de despliegue técnico sencillo, comenzando por una solución inmediata basada en GitHub Pages y evolución progresiva hacia infraestructuras más complejas si el proyecto lo demanda. Inicialmente, la aplicación puede publicarse de forma **gratuita, rápida y pública** gracias al uso de GitHub Pages, lo que permite compartir una demo funcional con pymes sin incurrir en gastos ni configuraciones complicadas. Se describió cómo organizar el repositorio, publicar el sitio estático y añadir una página de aterrizaje explicativa, asegurando que incluso usuarios sin formación técnica puedan comprender y acceder a la herramienta. Posteriormente, se discutieron posibles mejoras futuras: la adopción de un dominio propio para reforzar la imagen profesional, y la incorporación de un backend (en Node.js, PHP junto con una base de datos) para dotar al sistema de nuevas funcionalidades actualmente fuera del alcance de una SPA pura. También se analizaron las consideraciones de **escalabilidad** (tanto en número de usuarios como en complejidad de funcionalidades) y el **mantenimiento** en cada escenario, destacando el bajo mantenimiento de la solución estática inicial versus las mayores responsabilidades al introducir servidores y bases de datos. Finalmente, se brindaron recomendaciones orientadas a un despliegue profesional, cubriendo buenas prácticas como CI/CD, monitoreo, seguridad y soporte continuo.

Capítulo 11. CONCLUSIONES Y TRABAJOS

FUTUROS

11.1 RESUMEN DE LOGROS DEL PROYECTO

A lo largo del desarrollo del proyecto se ha concretado con éxito la idea inicial hasta alcanzar un prototipo funcional que cumple los objetivos planteados. Partimos de un análisis exhaustivo de requisitos y una revisión bibliográfica que orientó el diseño arquitectónico del sistema. En la fase de implementación, se seleccionaron herramientas tecnológicas adecuadas para el entorno industrial, integrando correctamente el backend con la interfaz de usuario y los componentes de hardware o software correspondientes. El prototipo final fue sometido a pruebas de validación donde demostró su eficacia en las tareas para las que fue concebido, mostrando un comportamiento estable y los resultados esperados en los diferentes escenarios planteados. En conjunto, este proceso permitió consolidar un producto que evidencia una mejora tangible respecto a soluciones previas, destacando el aprendizaje práctico obtenido desde la planificación hasta la puesta en marcha operativa.

11.2 LECCIONES APRENDIDAS

Durante la ejecución del proyecto se identificaron múltiples aprendizajes clave tanto en el ámbito técnico como organizativo:

Gestión del alcance y objetivos: Al iniciar el proyecto fue fundamental definir claramente los requerimientos y alcance; sin embargo, también surgieron necesidades adicionales en el camino. Se aprendió que un control estricto del alcance inicial ayuda a evitar desviaciones de tiempo y coste, aunque es necesario mantener cierta flexibilidad para

incorporar mejoras significativas. Este balance entre planificación rígida y adaptación permitió ajustar el rumbo sin comprometer la viabilidad del prototipo.

Desafíos técnicos y soluciones: En el ámbito técnico enfrentamos retos como la integración de distintos módulos de software y hardware, la selección de tecnologías compatibles y la optimización del rendimiento. Por ejemplo, al integrar la capa de persistencia de datos con el frontend surgieron incompatibilidades de formato que requerían transformaciones específicas. Estas dificultades enseñaron la importancia de una arquitectura modular y de documentar claramente las interfaces entre componentes. Cada obstáculo técnico abordado (depuración de errores, pruebas unitarias, optimización de consultas) fortaleció nuestras capacidades de resolución de problemas y reforzó buenas prácticas de desarrollo.

Coordinación y comunicación: La gestión del proyecto exigió mantener una comunicación fluida con tutores y colaboradores. La experiencia demostró que establecer canales claros de reporte de avances (reuniones periódicas, informes de estado) y asignar tareas con plazos definidos mejora la productividad. Se aprendió también la relevancia de la documentación continua: registrar las decisiones de diseño, los cambios en los requisitos y los resultados de las pruebas facilita la trazabilidad del proceso y reduce retrabajos posteriores. En resumen, la coordinación efectiva del equipo (incluso si el proyecto fue realizado de forma individual) es esencial para cumplir los hitos temporales y para resolver dudas de manera ágil.

Adaptación al entorno de trabajo: Durante el desarrollo se reforzó la capacidad de aprendizaje autónomo. Algunas tecnologías o metodologías utilizadas eran nuevas para nosotros, lo que implicó un proceso de autoformación (manipulación de nuevas librerías, lenguajes de programación o equipos de laboratorio). La experiencia mostró que dedicar tiempo a prototipos rápidos y a iteraciones tempranas permitió validar conceptos antes de invertir esfuerzos considerables. Asimismo, el manejo del tiempo entre trabajo de campo, laboratorio y estudio bibliográfico permitió mejorar la organización personal, equilibrando las tareas académicas con las responsabilidades del proyecto.

11.3 IMPLEMENTACIÓN FUTURA

A partir de la experiencia adquirida, se sugieren varias líneas de mejora que podrían ampliar la funcionalidad y robustez del sistema en desarrollos posteriores:

Tabla 2: Líneas de mejora

Área	Mejora propuesta
Arquitectura	Implementar un backend escalable mediante microservicios o arquitectura modular. Esto facilitaría la integración con nuevos servicios, la distribución de carga y el mantenimiento del código fuente. B3Se recomienda también documentar la API de comunicación entre módulos para simplificar futuras ampliaciones.
Seguridad	Incorporar mecanismos de autenticación y autorización más robustos, como protocolos OAuth2 o JWT para APIs, cifrado de datos en tránsito (HTTPS/TLS) y en reposo, y políticas de control de acceso. Además, sería recomendable realizar pruebas de penetración y auditorías periódicas para garantizar la protección contra vulnerabilidades.
Persistencia de datos	Ampliar la estrategia de almacenamiento hacia soluciones de base de datos empresarial (SQL/NoSQL) que aseguren la persistencia y consistencia de la información. Implementar backup automático y redundancia para evitar pérdida de datos. En proyectos futuros, explorar la integración con servicios en la nube (por ejemplo, AWS, Azure) para escalabilidad y disponibilidad.
Interfaz de usuario	Rediseñar la interfaz gráfica para mejorar la experiencia de usuario, poniendo énfasis en la usabilidad y accesibilidad. Esto incluye diseñar vistas responsive que se adapten a distintos dispositivos móviles, así como incorporar guías visuales intuitivas y feedback inmediato en interacciones críticas. También sería útil implementar tests de usabilidad con usuarios reales para validar la eficacia del diseño.
Pruebas y validación	Extender la cobertura de pruebas automatizadas (tests unitarios, de integración y de sistema) para detectar errores de forma temprana. Documentar casos de prueba y resultados para futuras referencias. También se podría establecer un entorno de pruebas continuo (CI/CD) que simplifique las compilaciones y despliegues, mejorando la calidad y rapidez en la entrega de actualizaciones.

Fuente: Elaboración propia

Además de estas mejoras, sería beneficioso considerar aspectos como la interoperabilidad con otros sistemas industriales (protocolos de comunicación estándares), la internacionalización del software (soporte multilenguaje) y la implementación de métricas de rendimiento para monitorizar el uso en tiempo real. Todas estas recomendaciones buscan ampliar el alcance del proyecto y asegurar su relevancia y adaptabilidad en aplicaciones reales o comerciales.

11.4 REFLEXIÓN FINAL

En calidad de autor de este trabajo, considero que este proyecto ha sido una experiencia decisiva en mi formación profesional. El proceso me ha impulsado a consolidar conocimientos técnicos en ingeniería, así como habilidades de gestión de proyectos y

trabajo autónomo. Me he enfrentado a situaciones desafiantes que requirieron paciencia y perseverancia, lo cual me permitió crecer tanto en competencias técnicas (como programación, diseño de sistemas y análisis de datos) como en soft skills (planificación, comunicación y adaptabilidad).

De manera honesta puedo decir que concluir este TFM ha reforzado mi motivación por los retos tecnológicos y me ha enseñado la importancia de la mejora continua. Este trabajo no solo representa el aprendizaje y esfuerzo invertidos, sino que también establece una base sólida para mi futuro profesional. En adelante, llevaré conmigo las lecciones aprendidas aquí (desde la importancia de la documentación rigurosa hasta el entusiasmo por explorar nuevas soluciones) para aplicarlas en proyectos industriales reales. Confío en que la experiencia acumulada enriquecerá mi contribución en el ámbito de la ingeniería en tecnologías industriales, ayudándome a enfrentar con éxito los retos profesionales que vengan.

Capítulo 12. BIBLIOGRAFÍA

- [1] Connolly, C. (2025, June 9). Using HTML and CSS to build a fast, modern website. *ProfileTree Web Design and Digital Marketing*.
<https://profiletree.com/using-html-and-css-to-build-a-fast-modern-website-without-javascript-overload/>
- [2] Again, M. C. S. G. (2024, November 28). Local Storage vs Session Storage Differences - Make Computer Science Great Again - Medium. *Medium*.
<https://medium.com/@MakeComputerScienceGreatAgain/local-storage-vs-session-storage-differences-5104765f5b24>
- [3] Lopera, J. D. P. (2025, May 13). CSRD y GRI: Principales Diferencias y Cómo Cumplir con Ambos Estándares en tu Estrategia ESG | Anthesis. *Anthesis*.
<https://www.anthesisgroup.com/co/art%C3%ADculos/csrd-y-gri-principales-diferencias-y-como-cumplir-con-ambos-estandares-en-tu-estrategia-esg/>
- [4] Dom. (2020, August 24). *Building a Single Page App without frameworks*. DEV Community. <https://dev.to/dcodeyt/building-a-single-page-app-without-frameworks-hl9>
- [5] Kinsta. (2025, February 28). *¿Qué Es un sitio Web Estático? Guía para Principiantes Absolutos - Kinsta®*. Kinsta®. <https://kinsta.com/es/base-de-conocimiento/que-es-un-sitio-web-estatico/>
- [6] Wolf Agencia de marketing digital. (2025, January 26). *Navegadores compatibles con HTML5 y su rendimiento en la web moderna*. Wolf Agencia De Marketing Digital. <https://wolfagenciademarketing.com/que-navegador-es-compatible-con-html5/>
- [7] Bridge. (2024, May 27). *Cómo hacer aplicaciones web escalables*. The Bridge | Digital Talent Accelerator. <https://thebridge.tech/blog/como-hacer-aplicaciones-web-escalables/>

- [8] Staniforth, J. (2025, May 13). *Key barriers: costs, complexity & lack of resources*. Sustainable Business Services. <https://sbs.eco/key-barriers-esg-carbon-reporting/>
- [9] Madera, M. (2023, March 11). *Cómo publicar tu sitio web con GitHub Pages*. Marco Madera: Comparto Mis Conocimientos, Pensamientos Y Opiniones Sobre La Programación. <https://marcomadera.com/blog/github-pages>
- [10] Primo, D. (2024, November 7). *Cómo organizar las carpetas de un proyecto de JavaScript en 8 pasos*. Web Reactiva. <https://www.webreactiva.com/blog/organizar-estructura-proyecto-incremental>
- [11] *Node.js: Desarrollo Backend de aplicaciones web*. EDI. (n.d.). FEVAL FORMACION. <https://formacionfeval.com/index.php/component/eventbooking/2024-34>
- [12] Carrillo, A. J. (2024, January 16). *CSRD: cómo transitar este nuevo paradigma* - KPMG Tendencias. KPMG Tendencias. <https://www.tendencias.kpmg.es/2023/07/csrd-como-transitar-este-nuevo-paradigma/>
- [13] ASOCIACIÓN INTERAMERICANA DE CONTABILIDAD. (2024, October 5). *La responsabilidad social empresarial en las PYMES: retos, oportunidades y aplicaciones prácticas* - AIC. AIC. <https://contadores-aic.org/la-responsabilidad-social-empresarial-en-las-pymes-retos-oportunidades-y-aplicaciones-practicas/>
- [14] Administrator, S. (2024, October 28). *PYMES y sostenibilidad: nuevos retos de información ESG*. Sygris. <https://sygris.com/2024/07/22/pymes-y-sostenibilidad-nuevos-retos-de-informacion-esg/>
- [15] amec Positive Industry. (2025, May 29). *Nueva herramienta para simplificar Informes de sostenibilidad en PYMEs* - amec Positive Industry. Amec Positive Industry. <https://www.amec.es/anticipate/alertas-y-oportunidades/nueva-herramienta-para-simplificar-informes-de-sostenibilidad-en-pymes/>
- [16] elEconomista.es. (2024, July 1). *Más del 80% de las pymes ve barreras a la hora de integrar la ESG*. elEconomista.es. <https://www.eleconomista.es/mercados->

- [cotizaciones/noticias/12890983/07/24/mas-del-80-de-las-pymes-ve-barreras-a-la-hora-de-integrar-la-esg.html](https://www.comillas.es/cotizaciones/noticias/12890983/07/24/mas-del-80-de-las-pymes-ve-barreras-a-la-hora-de-integrar-la-esg.html)
- [17] Business Conexión. (2024, October 11). Las 16 mejores herramientas de informes ESG en 2024. <https://www.businessconexion.info/2024/10/08/las-16-mejores-herramientas-de-informes-esg-en-2024/>
- [18] Calendario de Implementación de la Normativa ESG: CSRD en la Unión, Savills España. (2019). <https://www.savills.es/>
- [19] Carretero, L. M. (2021, March 22). APlanet, un estímulo tecnológico para dar brillo a la sostenibilidad. https://www.abc.es/economia/abci-aplanet-estimulo-tecnologico-para-brillo-sostenibilidad-202103220103_noticia.html
- [20] Castilla Vida, A., & Andreu Pinillos, A. (n.d.). VII Informe Comparativo de los Estados de Información NO Financiera (EINF) del Ibex-35. https://www.ey.com/es_es/insights/rethinking-sustainability/vii-informe-comparativo-estados-informacion-no-financiera-ibex-35
- [21] Cámara de España. (s.f.). Casi el 90 % de las empresas afirma percibir beneficios al aplicar medidas de sostenibilidad medioambiental. <https://www.camara.es/casi-90-empresas-afirma-percibir-beneficios-aplicar-medidas-sostenibilidad>
- [22] Comisión Europea. (2025, 26 de febrero). Corporate sustainability reporting. https://finance.ec.europa.eu/capital-markets-union-and-financial-markets/company-reporting-and-auditing/company-reporting/corporate-sustainability-reporting_en
- [23] Cuatro de cada diez pymes españolas incorporan criterios de sostenibilidad en su gestión. Cámara de España. (n.d.). <https://www.camara.es/cuatro-diez-pymes-espanolas-incorporan-criterios-sostenibilidad-gestion>
- [24] Datos 2023 de DGIPYME (Ministerio de Industria) citados en Club del Emprendimiento. Merino, N., & Merino, N. (2024, July 26). <https://www.clubdeemprendimiento.com/blog/pymes/analisis-de-la-evolucion-de-las-pymes-en-espana/>

- [25] Derrick, M. (2024, December 19). Explained: SAP's Green Ledger supports carbon tracking. <https://energydigital.com/technology-and-ai/sap-moving-carbon-data-from-averages-to-actuals>
- [26] Edwards, D. (2024, October 29). Freemium vs Free Trial: Which is Right for You | ProdPad. <https://www.prodpad.com/blog/freemium-vs-free-trial/>
- [27] Elisa Molero. (2023). Get ready to report: The ESG reporting software market will reach \$4.5 billion by 2027. <https://www.verdantix.com/insights/blogs/get-ready-to-report-the-esg-reporting-software-market-will-reach-4.5-billion-dollars-by-2027>
- [28] Finresp-Admin. (2024, November 12). Home - FinResP. <https://finresp.es/>
- [29] Freedman, R. (2021, November 4). SaaS companies quickly replacing subscriptions with usage-based pricing. <https://www.cfodive.com/news/saas-companies-quickly-replacing-subscriptions-with-usage-based-pricing/609497/>
- [30] Freedman, R. (2021, November 4). SaaS companies quickly replacing subscriptions with usage-based pricing. <https://www.cfodive.com/news/saas-companies-quickly-replacing-subscriptions-with-usage-based-pricing/609497/>
- [31] Gallo, J. (2021, 14 de diciembre). El Informe no Financiero, un aliado del largo plazo de la sostenibilidad. <https://www.soziabile.es/entorno/el-informe-no-financiero-un-aliado-del-largo-plazo-de-la-sostenibilidad>
- [32] Gil, L. (2024, September 2). How to calculate a B2B SaaS marketing budget. <https://www.effiqs.com/resources/b2b-saas-marketing-budget#:~:text=,Brand%20building%20and%20PR%3A%205%E2%80%9310>
- [33] Gomis, L. S. (2024, November 28). Los mejores Software ESG de 2025 para adoptar criterios de sostenibilidad en tu empresa. <https://www.softwaredoit.es/software-compliance/software-esg.html>
- [34] Gomis, L. S. (2024, November 28). Los mejores Software ESG de 2025 para adoptar criterios de sostenibilidad en tu empresa. <https://www.softwaredoit.es/software-compliance/software-esg.html>
- [35] Guide. (2025, February 28). M3ter. <https://www.m3ter.com/guides/usage-based-pricing>

- [36] HubSpot. (s.f.). B2B Buyers: Data, Trends, and Insights.
<https://blog.hubspot.com/sales/b2b-buyers#:~:text=social%20media,is%20marketing%20emails%20at%2041>
- [37] IT Digital Media Group. (2022, November 25). El 87 % de las empresas prevén invertir más en sostenibilidad los próximos dos años.
<https://www.ituser.es/actualidad/2022/11/el-87-de-las-empresas-preven-invertir-mas-en-sostenibilidad-los-proximos-dos-anos>
- [38] IT Digital Media Group. (2023, June 28). Banca, energía y aseguradoras son los sectores más implicados en asuntos ESG.
<https://sostenibilidad.ituser.es/actualidad/2023/07/banca-energia-y-aseguradoras-son-los-sectores-mas-implicados-en-asuntos-esg>
- [39] Jessen, J. (2024, July 6). Sweep leads global sustainability software rankings.
<https://sustainabilitymag.com/articles/sweep-leads-global-sustainability-software-rankings>
- [40] Juliane, & Juliane. (2025, February 19). Building a winning B2B SAAS Marketing & sales funnel. <https://mouseflow.com/blog/b2b-saas-marketing-funnel/>
- [41] Kaur, D. (2023, December 5). 30 case study statistics: Prove that it lead to more sales. <https://techstiks.com/case-study-statistics/>
- [42] KPMG International, Home. (2025, April 23). The move to mandatory reporting: Survey of Sustainability Reporting 2024. <https://kpmg.com/xx/en/our-insights/esg/the-move-to-mandatory-reporting.html>
- [43] MarketsandMarkets. (2023, 20 de marzo). The global ESG reporting software market to grow... <https://www.prnewswire.com/news-releases/the-global-esg-reporting-software-market-to-grow-from-usd-0-7-billion-in-2022-to-usd-1-5-billion-by-2027--at-a-compound-annual-growth-rate-cagr-of-15-9-301711792.html>
- [44] OpenView. (2023, March 3). Usage-Based Pricing: The next evolution in software pricing. <https://openviewpartners.com/usage-based-pricing/>

- [45] Pacto Mundial. (2023, June 12). Ley 11/2018 Información no financiera y diversidad. <https://www.pactomundial.org/leyes-directivas-normativas-sostenibilidad/ley-11-2018-de-28-de-diciembre-en-materia-de-informacion-no-financiera-y-diversidad/>
- [46] Pacto Mundial. (2024, 1 de agosto). Información no financiera - Oportunidades en el reporte de sostenibilidad. <https://www.pactomundial.org/noticia/diez-oportunidades-en-el-reporte-de-informacion-no-financiera/>
- [47] Plotkin, C. L., Stanley, J., & Harrison, L. (2024, September 12). Five fundamental truths: How B2B winners keep growing. <https://www.mckinsey.com/capabilities/growth-marketing-and-sales/our-insights/five-fundamental-truths-how-b2b-winners-keep-growing>
- [48] Precios de APLANET | Software ESG y Cumplimiento CSRD. (n.d.). <https://aplanet.org/es/precios/>
- [49] Puig-Serra, S. (2022, May 11). El 45% de las empresas no sabe que es obligatorio presentar su información no financiera. <https://www.grantthornton.es/sala-de-prensa/2021/el-45-de-las-empresas-medianas-no-sabe-que-es-obligatorio-presentar-un-informe-de-informacion-no-financiera/>
- [50] Reaching Beyond the Ready: How Thought Leadership Gets Out-of-Market B2B Buyers Back into the Game. (2024, February 28). <https://www.edelman.com/insights/thought-leadership-gets-b2b-buyers-back-into-game>
- [51] Redacción. (2025, 28 de enero). EcoVadis lanza una nueva herramienta para reportar sobre la CSRD. <https://www.soziabile.es/buen-gobierno/ecovadis-lanza-una-nueva-herramienta-para-reportar-sobre-la-csrd>
- [52] Salgado, L. (2023, December 20). Qué es la Doble Materialidad y Cómo Aplicarla. <https://aplanet.org/es/recursos/doble-materialidad/>
- [53] Salesforce Net Zero Cloud Pricing 2025. (n.d.). <https://www.g2.com/products/salesforce-net-zero-cloud/pricing>

- [54] Segal, M., & Segal, M. (2022, March 18). Google-Backed startup Normative launches free carbon calculator for SMEs. <https://www.esgtoday.com/google-backed-startup-normative-launches-free-carbon-calculator-for-smes/>
- [55] Soziable. (2024, febrero 28). La CSRD cuadruplicará el número de directores financieros con responsabilidades de sostenibilidad. <https://www.soziable.es/buen-gobierno/la-csrd-cuadruplicara-el-numero-de-directores-financieros-con-responsabilidades-de>
- [56] TICPymes, R., & TICPymes, R. (2023, December 13). Solo el 8% de las pymes informan de su impacto mediambiental. <https://www.ticpymes.es/noticias/solo-el-8-de-las-pymes-informan-de-su-impacto-mediambiental/>
- [57] Usage-based pricing (Consumption based pricing) Guide. (2025, February 28). <https://www.m3ter.com/guides/usage-based-pricing>