



**ZABALBURU**  
IKASTETXEA - S.COOP.

## **Sostenibilidad**

### **Reto 01: Recyclon Reloaded: El Desafío Digital**

#### **Autores:**

Pablo Santiago Clavijo Paredes

Asier Mateo Alvarez

Aaron David Marrero Vargas

David Duque Molina

Diego Castillo Macavilca

#### **Curso**

Primer año formativo de “DAW”

**Bilbao-España**

2025-2026

# **Índice**

Índice 1.-

INTRODUCCIÓN... (Nivel 1) .....	
1.1. Compromiso con la Sostenibilidad... (Nivel 2) .....	
1.2. Fase: Desarrollo (Nivel 2) .....	
1.2.1. Prácticas sostenibles... (Nivel 3) .....	
1.2.2. Prácticas no sostenible... (Nivel 3) .....	

# **Introducción del Proyecto: Transformación Digital Sostenible para Recyclon S.L.S.**

El presente proyecto aborda el "Reto 01: Recyclon Reloaded: El Desafío Digital", donde asumimos el rol de consultores clave para Recyclon S.L.S., una empresa especializada en la comercialización de material y mobiliario de oficina. Nuestro objetivo principal es liderar su transformación digital, debido a que su crecimiento exitoso ha dejado obsoletos sus procesos de gestión manuales y herramientas.

Nuestra misión es construir un ecosistema digital completo mediante el desarrollo de un portal web corporativo bilingüe que centralizará y optimizará las operaciones de la empresa.

## **Compromiso con la Sostenibilidad Digital.**

Para garantizar la integridad y la longevidad del proyecto, adoptaremos una metodología de trabajo profesional con un riguroso control de versiones en un repositorio de Github.

Más importante aún, implementaremos prácticas de sostenibilidad digital para crear un software eficiente y de baja huella energética, evitando el consumo innecesario de recursos técnicos y ambientales en el futuro. Esto incluye:

- **Código de Calidad:** Uso de código limpio, modular y reutilizable, lo que reduce la duplicación de funcionalidades y simplifica el mantenimiento.
- **Optimización de Recursos Web:** Aplicación de eco-diseño digital, como evitar animaciones innecesarias y optimizar recursos (imágenes comprimidas, scripts minimizados), para reducir el consumo de ancho de banda y la carga de CPU en los dispositivos.
- **Estabilidad Tecnológica:** Uso de frameworks y librerías estables y con soporte, asegurando que la solución no se vuelva obsoleta rápidamente, lo que evitaría la necesidad de rehacer el portal con un alto coste técnico y ambiental.

## **Fase: Desarrollo**

### **Prácticas sostenibles.-**

- Uso de código limpio, modular y reutilizable, reduciendo duplicación y facilitando el mantenimiento futuro sin necesidad de rehacer componentes.
- Optimización de recursos web (imágenes comprimidas, scripts minimizados, hojas de estilo eficientes) para disminuir el consumo de ancho de banda y energía en los servidores.
- Implementación de prácticas de eco-diseño digital, como evitar animaciones innecesarias que aumentan el uso de CPU en los dispositivos de los usuarios.
- Uso de frameworks y librerías estables que tengan comunidades activas y soporte, evitando rehacer funcionalidades y reduciendo el tiempo de desarrollo.
- Control de versiones en Git para evitar pérdidas de trabajo y asegurar un flujo ordenado sin generar redundancias o reprocesos.

### **Prácticas no sostenibles.-**

- Desarrollar sin una arquitectura clara, generando código espagueti que requerirá más tiempo, más energía de servidor y más recursos en el futuro.
- Cargar archivos multimedia en alta resolución sin optimización, aumentando el peso de la página y el consumo energético de los servidores al procesarla.
- Duplicar funcionalidades o código en lugar de crear componentes reutilizables, lo que incrementa el tiempo de desarrollo y mantenimiento.
- Usar frameworks obsoletos o no mantenidos, que obligarán a rehacer partes del portal y aumentarán el coste ambiental y técnico.
- No realizar pruebas de rendimiento durante el desarrollo, generando páginas lentas que demandan mayor consumo energético tanto del servidor como de los dispositivos clientes.

## **Estudio de Prácticas No sostenible/sostenibles:**

<b>Práctica no Sostenible : Desarrollar sin una arquitectura clara</b>	
<b>Identificación de la situación</b>	Se empieza a programar sin diseño previo ni estructura definida. Cada desarrollador organiza el código a su manera, generando un proyecto desordenado.
<b>Evaluación del impacto</b>	El código se vuelve difícil de mantener, aparecen más errores y retrasos, y aumenta la deuda técnica y la dependencia del conocimiento individual.
<b>Identificación de causas</b>	Falta de planificación, poca experiencia en diseño, ausencia de estándares y mala comunicación dentro del equipo.
<b>Práctica Sostenible: Uso de código limpio, modular y reutilizable</b>	
<b>Propuestas de mejora</b>	Definir una arquitectura clara, aplicar estándares de codificación y dividir el proyecto en módulos reutilizables, además de hacer revisiones de código.
<b>Plan de acción</b>	Crear la arquitectura base, documentar reglas internas, usar un flujo de trabajo ordenado en Git y realizar revisiones periódicas para asegurar calidad y coherencia.

**Nota:** tabla realizada por autores del reto 01. (EQUIPO 03)

<b>Práctica no Sostenible : Cargar archivos multimedia en alta resolución sin optimización</b>	
<b>Identificación de la situación</b>	Se suben imágenes y vídeos muy pesados directamente a la web sin reducir su tamaño, generando archivos innecesariamente grandes.
<b>Evaluación del impacto</b>	La página tarda más en cargar, afecta la experiencia del usuario, aumenta el consumo de datos y eleva el uso de recursos del servidor.
<b>Identificación de causas</b>	Falta de conocimiento sobre optimización, ausencia de estándares internos y uso directo del archivo original sin herramientas de compresión.
<b>Práctica Sostenible: Optimización de recursos web</b>	
<b>Propuestas de mejora</b>	Comprimir imágenes y vídeos, usar formatos eficientes (WebP, AVIF) y aplicar carga diferida (lazy load) para mejorar el rendimiento del sitio.
<b>Plan de acción</b>	Definir un protocolo de optimización, implementar herramientas automáticas (TinyPNG, Squoosh, scripts) y revisar periódicamente el peso de los recursos web.

**Nota:** tabla realizada por autores del reto 01. (EQUIPO 03)

<b>Práctica no Sostenible : Duplicar funcionalidades o código</b>	
<b>Identificación de la situación</b>	Se copian partes de código o funcionalidades completas en diferentes archivos o módulos, generando duplicados difíciles de mantener.
<b>Evaluación del impacto</b>	Aumenta la complejidad del proyecto, se multiplican los errores y es más costoso corregir fallos porque deben actualizarse en varios lugares.
<b>Identificación de causas</b>	Falta de organización, desconocimiento de patrones reutilizables y presión por soluciones rápidas sin revisar si ya existe código similar.
<b>Práctica Sostenible: Implementación de prácticas de eco-diseño digital</b>	
<b>Propuestas de mejora</b>	Centralizar funciones reutilizables, aplicar patrones de diseño, usar librerías compartidas y fomentar una arquitectura que evite repetir lógica.
<b>Plan de acción</b>	Analizar el código existente, crear módulos reutilizables, documentar componentes comunes y establecer revisiones periódicas para evitar duplicaciones.

**Nota:** tabla realizada por autores del reto 01. (EQUIPO 03)

<b>Práctica no Sostenible : Usar frameworks obsoletos o no mantenidos</b>	
<b>Identificación de la situación</b>	El proyecto utiliza frameworks antiguos sin actualizaciones, seguridad activa ni soporte oficial, lo que limita nuevas funcionalidades y compatibilidad.
<b>Evaluación del impacto</b>	Aumenta el riesgo de vulnerabilidades, se dificulta integrar tecnologías modernas y se incrementan los costes de mantenimiento y tiempo de desarrollo.
<b>Identificación de causas</b>	Falta de actualización técnica, desconocimiento de alternativas actuales o resistencia al cambio por parte del equipo.
<b>Práctica Sostenible: Uso de frameworks y librerías estables</b>	
<b>Propuestas de mejora</b>	Actualizar a versiones soportadas, evaluar librerías con buena comunidad activa y mantenerse atento a novedades con ciclos de actualización periódicos.
<b>Plan de acción</b>	Auditar los frameworks del proyecto, seleccionar alternativas modernas, planificar la migración por fases y documentar la nueva estructura tecnológica.

**Nota:** tabla realizada por autores del reto 01. (EQUIPO 03)

<b>Práctica no Sostenible : No realizar pruebas de rendimiento</b>	
<b>Identificación de la situación</b>	La aplicación se desarrolla y despliega sin evaluar tiempos de carga, uso de recursos o comportamiento bajo carga, dejando posibles problemas ocultos.
<b>Evaluación del impacto</b>	La web puede volverse lenta, inestable o colapsar con muchos usuarios; empeora la experiencia del cliente y aumenta el tiempo de corrección tras el fallo.
<b>Identificación de causas</b>	Falta de herramientas o conocimientos, poca planificación de pruebas o presión por entregar rápido sin validar el rendimiento.
<b>Práctica Sostenible: Control de versiones en Git</b>	
<b>Propuesta de mejora</b>	Implementar flujos de trabajo como GitFlow, realizar commits frecuentes y claros, y usar ramas para pruebas, nuevas funciones y correcciones sin afectar el código principal.
<b>Plan de acción</b>	Configurar un repositorio bien estructurado, definir normas internas de uso de Git, realizar revisiones de código y hacer integraciones periódicas para evitar conflictos.

**Nota:** tabla realizada por autores del reto 01. (EQUIPO 03)