

Guía de Proyecto Web - SportEco

Guía de Proyecto Web - SportEco
Tienda de material deportivo sostenible

1. Descripción general del proyecto

SportEco es una tienda web de material deportivo sostenible, centrada en productos fabricados con materiales reciclados o de proximidad. El proyecto se alinea con los Objetivos de Desarrollo Sostenible (ODS) 3 (Salud y bienestar) y 12 (Producción y consumo responsables). La web contará con 4 páginas principales:

1. index.html: página de inicio con campañas y acceso al resto de páginas.
2. productos.html: catálogo de material deportivo sostenible.
3. compra.html: configuración y compra de packs de material sostenible.
4. contacto.html: página de soporte y perfil deportivo del usuario.

Tecnologías principales:

- HTML5, CSS3, JavaScript (cliente)
- Node.js + Express (servidor)
- API externa (por ejemplo, datos de emisiones/huella o clima para actividades deportivas)
- Git y GitHub para control de versiones

Cada integrante tendrá una carga de trabajo similar: HTML + CSS + JavaScript en cliente, integración con el servidor y referencias claras a los ODS.

2. Estructura general de la web

Todas las páginas compartirán:

- Cabecera común con logo SportEco, menú de navegación y pequeño recordatorio de los ODS.
- Pie de página con enlaces rápidos, créditos y mención a ODS 3 y 12.
- Archivo CSS base (por ejemplo, css/base.css) para tipografías, colores corporativos y layout general.

Páginas:

- index.html
- productos.html
- compra.html
- contacto.html

3. Reparto de tareas por integrante

Integrante 1 – index.html (campañas + ODS + accesibilidad)

Responsabilidades principales:

- Crear index.html con:
 - Cabecera, menú de navegación accesible (enlace claro a productos, compra y contacto).
 - Banner principal accesible (texto alternativo, contraste correcto, sin contenido esencial solo en la imagen).
 - Sección de campañas de material sostenible (correr, gym, montaña, etc.).
 - Módulo donde se explique explícitamente la relación de SportEco con los ODS 3 y 12.
- CSS específico (por ejemplo css/index.css):

- Diseño adaptable (layout responsive sencillo).
- Estilos para el banner, tarjetas de campañas y sección ODS.
- JavaScript (por ejemplo js/index.js):
 - Pequeño slider o carrusel de campañas (automático o con botones “anterior/siguiente”).
 - Mostrar/ocultar cuadro informativo sobre ODS al hacer clic en un botón (“¿Por qué somos sostenibles?”).
 - Cambios de estilo al interactuar (botones activos, foco accesible, etc.).
- Integración con Node.js:
 - Definir una ruta GET en el servidor (por ejemplo /campanias) que devuelva un JSON con campañas.
 - Hacer una petición fetch desde index.js para cargar dinámicamente las campañas (títulos, descripción, tipo de deporte).
 - Opcional: permitir que las campañas se definan en un archivo JSON que el servidor sirva.

ODS en esta página:

- Explicar cómo el deporte contribuye a la salud (ODS 3).
- Mostrar mensajes sobre la importancia de elegir equipamiento sostenible (ODS 12).

Integrante 2 – productos.html (catálogo de productos)

Responsabilidades principales:

- Crear productos.html con:
 - Listado de productos en tarjetas (zapatillas, camisetas, botellas reutilizables, etc.).
 - Filtros por categoría (correr/gym/montaña), por tipo de material (reciclado, orgánico, local) y por precio.
 - Iconos o etiquetas en las tarjetas para indicar impacto (baja huella, kilómetro cero...).
- CSS específico (css/productos.css):
 - Diseño de un grid de productos (flex o grid).
 - Estilos para etiquetas de sostenibilidad (colores consistentes con los ODS).
- JavaScript (js/productos.js):
 - Carga dinámica de productos desde el servidor (fetch a /productos).
 - Filtros en vivo: al seleccionar un filtro se actualiza el listado sin recargar la página.
 - Cambios de estilo: resaltar productos “top sostenibles” o con descuentos verdes.
 - Posibilidad de marcar productos como favoritos (ícono de corazón, por ejemplo) y almacenarlo temporalmente (localStorage).
- Integración con Node.js:
 - Ruta GET /productos que devuelva un JSON con la lista de productos y sus propiedades (nombre, precio, categoría, etiquetas de sostenibilidad).
 - Opcional: poder filtrar en servidor o dejar el filtrado al cliente.

ODS en esta página:

- Mostrar claramente qué significa que un producto es “sostenible” (material reciclado, local, durabilidad).
- Posible contador o mensaje de CO2 estimado ahorrado por producto respecto a alternativas convencionales.

Integrante 3 – compra.html (packs y compra de material sostenible)

Responsabilidades principales:

- Crear compra.html con:
 - Un área de “catálogo reducido” para formar un pack de inicio (zapatillas + camiseta + botella + accesorios).
 - Zona “Mi pack SportEco” donde el usuario arrastra y suelta productos (drag&drop) desde el catálogo reducido.
 - Resumen del pack con precio total, descuento y resumen de impacto sostenible.
- CSS específico (css/compra.css):
 - Estilos para las zonas de drag&drop (catálogo reducido y pack).
 - Indicar visualmente qué elementos son arrastrables y dónde se pueden soltar.
- JavaScript (js/compra.js):
 - Implementar drag&drop: el usuario arrastra un producto del catálogo al área “Mi pack”.
 - Cálculo dinámico de precio total.
 - Lógica de descuento: si el usuario combina ciertos productos sostenibles (por ejemplo, zapatillas + camiseta reciclada + botella reutilizable), aplicar un descuento extra “eco-pack”.
 - Muestra adicional del impacto sostenido del pack (por ejemplo, número de productos reciclados, mensaje motivacional).
 - Validación básica de un pequeño formulario de compra (nombre, email, método de pago simulado).
- Integración con Node.js:
 - Ruta POST /compra que reciba los datos del pack (lista de productos, precio total, descuento, datos del cliente).
 - Guardar el pedido en memoria o en un archivo JSON (simulación de base de datos).
 - Devolver una respuesta en JSON con un mensaje de confirmación para mostrar en la página (sin recargar).

ODS en esta página:

- Destacar que los packs fomentan el consumo responsable (comprar una vez, productos duraderos y sostenibles).
- Mensaje sobre la salud y el bienestar asociados a la práctica deportiva (ODS 3).

Integrante 4 – contacto.html (contacto, soporte y perfil deportivo)

Responsabilidades principales:

- Crear contacto.html con:
 - Formulario de contacto y soporte: nombre, email, tema (duda sobre producto, problema con pedido, sugerencia), mensaje.
 - Formulario de perfil deportivo: tipo de deporte principal, nivel (principiante/intermedio/avanzado), frecuencia semanal, objetivos (salud, rendimiento, bajar huella, etc.).
 - Cuadros de texto o bloques que expliquen cómo SportEco apoya el deporte saludable y sostenible.
- CSS específico (css/contacto.css):
 - Estilos de formularios en columna usando flex.
 - Destacar campos obligatorios e incluir mensajes de error legibles.
- JavaScript (js/contacto.js):
 - Validación de formularios en el lado cliente: campos obligatorios, formato de email,

longitud mínima de mensaje.

- Creación dinámica de mensajes de error (<p class="error">) debajo o encima del formulario.

- Mostrar un mensaje de éxito sin recargar la página cuando el formulario se envía correctamente.

- Opcional: generar un pequeño “resumen de perfil” a partir de las respuestas del usuario y mostrar recomendaciones de página del sitio.

- Integración con Node.js:

- Ruta POST /contacto para recibir consultas y guardarlas en memoria/archivo JSON (con fecha, tipo de consulta, email, etc.).

- Ruta POST /perfil para guardar el perfil deportivo del usuario (similar a un registro de preferencias).

- Respuestas JSON para confirmar recepción y permitir feedback instantáneo en la interfaz.

ODS en esta página:

- Preguntas orientadas a la salud (motivaciones, bienestar, deporte regular) que refuercen el ODS 3.

- Mensajes que animen a elegir equipamiento que reduzca el impacto ambiental (ODS 12).

4. Tareas comunes y coordinación

Además del trabajo específico de cada página, el equipo deberá coordinar:

- Definir un diseño común (colores, tipografía, estilos de botones) coherente con la idea de sostenibilidad.

- Crear y mantener un repositorio GitHub compartido.

- Dividir y documentar el código de Node.js (app.js, rutas de cada integrante, estructura de carpetas).

- Revisar que todas las páginas sean accesibles (texto alternativo, contraste, estructura semántica).

- Incluir una sección (puede ser en index.html o en una subpágina) donde se explique claramente la relación del proyecto con los ODS.

- Preparar una breve documentación o memoria técnica basada en esta guía para la entrega final.

Con este reparto, cada integrante:

- Diseña y maqueta una página HTML completa.

- Crea CSS específico para su página, además de usar el CSS común.

- Programa funcionalidad JavaScript visible para el usuario.

- Integra al menos una ruta del servidor en Node.js.

- Incluye mensajes o elementos que conecten con los ODS 3 y 12.

Esto asegura una carga de trabajo equilibrada y un proyecto coherente y completo, orientado a obtener la máxima nota posible.