

Sobriedad digital

M1: Orientación al perfil de especialidades y metodología del curso

AE: Proponer prácticas de sustentabilidad utilizando el concepto de sobriedad digital en el contexto de la industria de la tecnología.



Introducción



La aceleración del consumo digital — cada vídeo en streaming, cada línea de código desplegada en la nube, cada dispositivo fabricado — ha colocado a las TIC entre los sectores con mayor crecimiento en su huella de carbono. Frente a este escenario surge la **sobriedad digital**, un enfoque que invita a repensar nuestras necesidades tecnológicas y a diseñar soluciones que maximicen valor con el mínimo de recursos energéticos y materiales.

La sobriedad digital no consiste en frenar la innovación, sino en encauzarla: cuestiona cuándo, dónde y cómo usamos la tecnología para que sus beneficios no se vean opacados por impactos sociales y ambientales adversos. Al adoptar esta perspectiva, las organizaciones pueden reducir costes operativos, mejorar la accesibilidad de sus productos y, al mismo tiempo, contribuir a los Objetivos de Desarrollo Sostenible (ODS).

Aprendizaje esperado

Cuando finalices la lección serás capaz de:

- Proponer prácticas de sustentabilidad utilizando el concepto de sobriedad digital en el contexto de la industria de la tecnología.

¿Qué es la sobriedad digital?

Definición y alcance

La sobriedad digital es la estrategia de limitar el consumo de energía, datos y hardware a lo estrictamente necesario para alcanzar los objetivos deseados. Su alcance abarca todo el ecosistema digital: infraestructura, software, dispositivos y hábitos de uso. A diferencia de enfoques centrados solo en la eficiencia, la sobriedad digital también plantea la pertinencia de ciertas funcionalidades o volúmenes de datos (no producir, no almacenar, no transmitir cuando no aporta valor).

Concepto	Enfoque principal	Pregunta guía
Green IT	Eficiencia de sistemas existentes	«¿Cómo hago lo mismo con menos recursos?»
Sobriedad digital	Pertinencia y suficiencia	«¿Es necesario hacer esto? ¿Hay una forma más sencilla o ligera?»

Importancia de la sobriedad digital en el contexto actual de la sociedad

- **Ambiental:** El sector digital supera ya a la aviación comercial en emisiones de GEI. Reducir transferencia de datos y prolongar la vida útil del hardware es crucial.
- **Social:** La hiperconectividad puede profundizar la fatiga digital y las brechas de acceso; diseñar experiencias ligeras mejora la inclusión.
- **Económica:** Menos recursos implican menos OPEX y CAPEX, mayor resiliencia ante cambios de mercado y normativas ambientales.

Sobriedad digital y sostenibilidad

La sostenibilidad integra las dimensiones ambiental, social y económica. La sobriedad digital actúa como catalizador al fijar objetivos de consumo (energía, datos, materiales) y métricas de equidad (accesibilidad, privacidad) que guían las decisiones técnicas y de negocio.

Sobriedad digital en el desarrollo de software

Principios de diseño

- **Eficiencia:** Algoritmos de baja complejidad; uso de lazy loading, compresión y caché consciente.
- **Sostenibilidad:** Selección de lenguajes y runtimes con menor overhead; despliegues en centros de datos con energía renovable.
- **Accesibilidad:** Interfaces ligeras compatibles con dispositivos de gama baja y conexiones lentas.
- **Privacidad:** Recopilación mínima de datos personales; cifrado de extremo a extremo.
- **Seguridad:** Principio de mínimo privilegio y pruebas continuas para evitar vulnerabilidades que generen tráfico innecesario (p. ej., bots).

Prácticas para aplicar la sobriedad digital en el ciclo de vida de desarrollo de software

Fase SDLC	Acción de sobriedad	Herramienta / Métrica sugerida
Planificación	Definir “presupuesto de carbono” por característica	Cuadro de mando de CO ₂ eq
Diseño	Arquitectura serverless o contenedores con auto-scaling inverso	Diagrama C4 + cálculo de carga pico
Implementación	Medir consumo energético en CI/CD	Plug-ins de PowerAPI, Scaphandre
Pruebas	Pruebas de rendimiento + impacto energético	Greenspector, Fruggr
Despliegue	Ubicar cargas en franjas horarias con baja intensidad de carbono	Carbon Aware SDK
Mantenimiento	Refactor para eliminar deuda técnica y código muerto	Índice de complejidad ciclomática

Durante la Codificación (Backend/Algoritmos)

- **Optimización de Consultas:** Evitar consultas ineficientes a la base de datos (Ej. usar índices, evitar **SELECT ***).
- **Algoritmos Eficientes:** Elegir algoritmos con la menor complejidad temporal (**O(n)**, **O(n)**).
- **Control de Logs:** Limitar la generación de logs innecesarios, que consumen espacio en disco y **CPU** de escritura.

Durante la Codificación (Frontend/UX)

- **Imágenes y Media:** Usar formatos de imagen modernos (**WebP**) y optimizar su peso. Cargar elementos solo cuando son visibles (**Lazy Loading**).
- **Unidades Relativas:** Usar unidades relativas (**rem**, **vw**) y **CSS** puro sobre grandes bibliotecas **JS** cuando sea posible.
- **Mobile First:** Priorizar la carga rápida y la experiencia en móviles.

Durante el Despliegue e Infraestructura

- **Uso de Caché:** Implementar caching agresivo en el lado del servidor y del cliente para reducir peticiones repetidas.
- **Servidores Eficientes:** Elegir proveedores de nube que utilicen energías renovables y usar contenedores optimizados (Ej. Alpine Linux).
- **Apagado Automático:** Configurar ambientes de desarrollo y staging para que se apaguen automáticamente en horas no laborales.

Ejemplo de Código y Arquitectura Ligera

El código simple y enfocado es más rápido, consume menos **CPU** y es más fácil de mantener.

Mala Práctica (Python): Carga todo en memoria y consume más **RAM**.

```
# Lectura de un archivo grande de datos (Ej. 5 GB)
datos_completos = open('logs_grandes.txt', 'r').readlines()
for linea in datos_completos:
    # Procesar linea
    ...

```

Buena Práctica (Python): Usa un generador para procesar línea por línea, minimizando la carga en RAM.

```
# Lectura de un archivo grande linea por linea
with open('logs_grandes.txt', 'r') as archivo:
    for linea in archivo:
        # Procesar linea
        # La RAM solo retiene una linea a la vez.
        ...

```

Arquitectura Ligera: Preferir arquitecturas **sin servidor (Serverless)** como AWS Lambda o Google Cloud Functions. Estas solo consumen energía y recursos cuando se ejecutan (event-driven), lo opuesto a un servidor EC2 que está encendido 24/7.

Herramientas y recursos para aplicar sobriedad digital en el desarrollo de software

- **Cloud Carbon Footprint:** estima emisiones en AWS, Azure y GCP.
- **EcoIndex / Ecograder:** analiza peso y eficiencia de páginas web.
- **OpenLCA:** modela Análisis de Ciclo de Vida (ACV) para servicios digitales.
- **GSF Carbon Aware SDK:** ajusta cargas a momentos con energía más limpia.

Impacto y responsabilidad

Evaluación del impacto ambiental y social de las decisiones tecnológicas

La evaluación integral contempla:

1. **Alcance 1 – 3** del GHG Protocol para emisiones directas e indirectas.
2. Indicadores sociales: accesibilidad, protección de datos, bienestar laboral.
3. Transparencia y reporte público de avances.

Análisis del ciclo de vida de los productos y servicios de software

El ACV digital considera cuatro etapas: fabricación de hardware, uso, fin de vida y reciclaje. Puntos clave:

- **Fabricación:** extracción de minerales críticos; alto impacto inicial.
- **Uso:** consumo eléctrico en centros de datos y dispositivos del usuario.
- **Fin de vida:** gestión de residuos electrónicos; reciclaje y reutilización.

Cálculo de la Huella de Carbono del Software

La huella de carbono (CO₂e) de un software depende de:

1. **Energía Consumida:** Cuánta electricidad usa el CPU durante el procesamiento.
2. **Mix Energético:** Si el centro de datos usa energía renovable o fósil.

Actividad Práctica Sugerida:

- **Herramientas:** Utilizar bibliotecas como **CodeCarbon (Python)** o **Cloud Carbon Footprint** para medir la energía y las emisiones de CO₂ de un algoritmo o una aplicación en la nube.
- **Ejemplo:** Medir las emisiones al ejecutar un ciclo for ineficiente vs. una versión optimizada. La diferencia de consumo energético cuantifica la ganancia en SD.

Análisis del Ciclo de Vida de los Productos y Servicios de Software

Debemos considerar el impacto en todas las etapas:

- **Fabricación (Hardware):** Uso de materiales raros y energía.
- **Uso (Software):** Consumo de energía del servidor y del dispositivo del usuario.
- **Fin de Vida:** Desechar el hardware o extender su utilidad.

Consideraciones éticas y sociales en el desarrollo de software

- Evitar dark patterns que incentiven consumo excesivo de datos o hardware.
- Diseñar opciones de “modo bajo consumo” para usuarios.
- Garantizar que optimizar recursos no comprometa la accesibilidad de personas con discapacidad.

Responsabilidad de los desarrolladores en la construcción de un futuro tecnológico sostenible

El rol profesional se amplía: de “creadores de código” a “guardianes del carbono”.

Recomendaciones:

- Incluir métricas de energía y CO₂ en la Definition of Done.
- Promover estándares abiertos y formación interna sobre diseño sostenible.
- Participar en comunidades y proyectos de código abierto orientados a la eficiencia.

Casos de estudio

Ejemplos reales en la industria tecnológica

1. **Worldline (2024-2025):** migración de centros de datos a nubes regionales alimentadas con 90 % de renovables → reducción del 42 % de CO₂eq en dos años.
2. **The Shift Project – STERM Model:** optimización de una app de mensajería que ahorró 30 % de energía ajustando compresión y caché.
3. **Fruggr:** integró una “puntuación de sobriedad” en su pipeline DevSecOps → tiempo de carga –22 %, consumo energético –18 %.

Análisis de casos de estudio para la evaluación del impacto ambiental de decisiones tecnológicas

Para cada caso se pueden identificar:

- **Variables de decisión:** ubicación de servidores, arquitectura, políticas de caché.
- **Indicadores medidos:** kWh, gCO₂eq, latencia, satisfacción de usuario.
- **Lecciones aprendidas:** importancia de medir desde el diseño; necesidad de reportar resultados para fomentar la mejora continua.

Actividades prácticas aplicadas al trabajo del estudiante

Actividad 1 – Medir el peso y la eficiencia de una página del proyecto

1. Elige una página de tu proyecto (landing, dashboard, formulario).
2. Abre las DevTools del navegador → pestaña **Network**.
3. Recarga la página y anota:

- Peso total transferido (MB).
 - Número de peticiones.
 - Tiempo de carga hasta DOMContentLoaded o Load.
4. Identifica los elementos más pesados (imágenes, scripts, fuentes).
 5. Propón al menos **tres mejoras** (por ejemplo, cambio de formato de imagen, eliminación de scripts no usados, carga diferida).
 6. Aplica las mejoras y repite la medición → compara “antes vs. después”.

Actividad 2 – Medir eficiencia de código (conceptual con CodeCarbon)

1. Elige un algoritmo de tu proyecto (por ejemplo, procesamiento de una lista grande).
2. Implementa una versión “naive” (poco optimizada).
3. Implementa una versión optimizada (mejor algoritmo, menos estructuras temporales).
4. Usa una biblioteca como **CodeCarbon** (o simula el uso leyendo la documentación) para:
 - Medir el tiempo de ejecución de cada versión.
 - Estimar el consumo energético y las emisiones (CO₂eq).
5. Registra resultados y arma una tabla comparativa (tiempo, energía, emisiones, complejidad).

Checklist de Sobriedad Digital

Esta checklist sirve como guía para desarrolladores y arquitectos en el SDLC.

Fase	Práctica de Sobriedad Digital (SD)	<input checked="" type="checkbox"/>
Diseño	¿Se aplicó el principio de "hacer menos, pero mejor"? ¿Se consideró el hardware de gama baja?	<input type="checkbox"/>
Backend	¿Están optimizadas las consultas a la DB? ¿Se limitaron los logs?	<input type="checkbox"/>
Frontend	¿El sitio es usable sin JavaScript? ¿Se usaron formatos de imagen modernos (WebP)?	<input type="checkbox"/>

Despliegue	¿Se está usando caching agresivo? ¿El servidor se apaga automáticamente en inactividad?	<input type="checkbox"/>
Métricas	¿Se están midiendo y reportando el consumo de CPU y transferencia de datos?	<input type="checkbox"/>

Dinámica: selecciona un caso para desarrollar y responder a las preguntas propuestas.

Caso de Estudio	Descripción	Preguntas de Análisis
Netflix/YouTube	Plataformas que ofrecen distintos niveles de calidad de video (SD, HD, 4K) y autoplay.	¿Qué oportunidades hay para la SD (Ej. desactivar autoplay por defecto, limitar la resolución máxima en móviles)? ¿Cuál es el desafío económico?
Aplicaciones Móviles "Light"	Ej. Facebook Lite, Twitter Lite.	¿Cómo estas versiones demuestran el principio de "Eficiencia y Accesibilidad"? ¿Qué funcionalidades esenciales sacrificaron y fue justificable?
Minería de Criptomonedas	Uso intensivo de hardware para el cálculo.	¿Cuál es el impacto ambiental y social directo de este modelo? ¿Qué consideraciones éticas deben tener los desarrolladores de blockchain?

Guía rápida – Uso de EcolIndex para una página web

1. Asegúrate de que tu página sea accesible desde una URL pública o un entorno accesible.
2. Entra al sitio de **EcolIndex**.
3. Ingresa la URL de tu página.
4. Espera el análisis y revisa:
 - Nota global (A–G).
 - Peso de la página.
 - Número de peticiones.
 - Tiempo de carga.
5. Toma notas de los indicadores más críticos.
6. A partir del reporte, define **3 acciones de mejora** (por ejemplo, reducir tamaño de imágenes, minimizar CSS/JS, implementar caché).

- Vuelve a analizar después de los cambios y compara resultados.

Guía rápida – Uso de Cloud Carbon Footprint (visión general)

- Identifica el proveedor de nube que usas (AWS, Azure, GCP).
- Revisa la documentación de **Cloud Carbon Footprint** para conectar tu cuenta de nube (requiere permisos de lectura de facturación/uso).
- Configura el proyecto para leer datos de consumo (horas de CPU, almacenamiento, tráfico).
- Genera un primer reporte de emisiones por:
 - Servicio (compute, storage, networking).
 - Región.
 - Proyecto o etiqueta.
- Analiza qué servicios o regiones concentran mayor huella.
- Propón decisiones de sobriedad digital basadas en el reporte:
 - Mover cargas a regiones con mejor mix renovable.
 - Apagar recursos infrautilizados.
 - Ajustar tamaños de instancias.

Modelos de comparación “antes vs. después” en eficiencia digital

Ejemplo 1 – Página web

Métrica	Antes optimización	Después optimización
Peso total	3.8 MB	1.4 MB
Número de peticiones	95	52
Tiempo de carga (3G)	6.2 s	2.9 s
EcoIndex (nota)	D	B

Acciones realizadas:

- Conversión de PNG → WebP.

- Eliminación de fuentes no utilizadas.
- Minificación de CSS y JS.

Ejemplo 2 – Algoritmo de procesamiento

Métrica	Versión naive	Versión optimizada
Complejidad estimada	$O(n^2)$	$O(n \log n)$
Tiempo ($n = 100.000$)	4.5 s	0.6 s
Consumo estimado CPU	Alto	Medio/Bajo
Emisiones CO ₂ (ejecución)	1.0× (referencia)	~0.2× (~80 %)

Estos modelos ayudan a **visualizar el impacto concreto** de las prácticas de sobriedad digital en términos de rendimiento, experiencia de usuario y huella ambiental.

Ejercicios grupales de evaluación ética y ambiental

Dinámica colaborativa general

1. Selecciona un caso de estudio.
2. Responde preguntas de análisis ético y ambiental.

Casos sugeridos

Caso	Descripción breve	Preguntas clave de análisis
Netflix / YouTube	Plataformas con autoplay, distintas calidades de vídeo.	¿Desactivar autoplay por defecto? ¿Límite de resolución en móvil? ¿Impacto económico vs. ambiental?
Aplicaciones “Lite”	Versiones ligeras (Facebook Lite, Twitter Lite, etc.).	¿Qué funcionalidades se sacrifican? ¿Mejora de accesibilidad y huella?
Minería de criptomonedas	Uso masivo de hardware para cálculo intensivo.	¿Impacto directo en energía y comunidades? ¿Qué dilemas éticos se presentan?

Plantilla para proponer prácticas de sobriedad digital en proyectos del bootcamp

Puedes utilizar esta estructura para documentar propuestas concretas en tus proyectos:

1. Contexto del proyecto

- Nombre del proyecto.
- Tipo de aplicación (web, API, móvil, dashboard, etc.).

2. Problema / oportunidad de sobriedad digital

- ¿Qué comportamiento, componente o decisión genera alto consumo o impacto?
- Evidencia: métricas actuales (peso, tiempo de carga, consumo estimado, etc.).

3. Objetivo de mejora

- Meta cuantitativa (por ejemplo: reducir peso de la página en un 40 %, bajar tiempo de carga por debajo de 3 s, disminuir llamadas a una API en un 30 %).

4. Propuestas de acción

- Acción 1 (descripción + responsable + impacto estimado).
- Acción 2...
- Acción 3...

5. Criterios de evaluación ambiental y social

- Indicadores ambientales (kWh, gCO₂eq, peso de página, uso de CPU).
- Indicadores sociales (accesibilidad, inclusión de dispositivos de gama baja, privacidad).

6. Riesgos y trade-offs

- ¿Qué se podría perder o dificultar al aplicar la mejora? (ej.: menos animaciones, cambio de formato de vídeo).
- ¿Cómo se mitigará?

7. Plan de seguimiento

- Frecuencia de mediciones.
- Cómo se reportarán los resultados al equipo.

Cierre

La sobriedad digital no es una moda pasajera, sino la evolución natural de un sector que aspira a ser parte de la solución climática. Cada línea de código, cada arquitectura y cada elección de infraestructura representan una oportunidad para reducir huella de carbono, mejorar la inclusividad y optimizar costos. Adoptar estos principios exige voluntad y metodología, pero los beneficios — tangibles e intangibles — superan con creces el esfuerzo inicial. Que este manual sirva como punto de partida para integrar la sobriedad digital en tu día a día profesional y promover un futuro tecnológico verdaderamente sostenible.

Referencias



- Green Software Foundation. (2025). Manifesto.
<https://greensoftware.foundation/manifesto/>
- The Shift Project. (2024). Deploying digital sobriety.
<https://theshiftproject.org/en/publications/deploying-digital-sobriety/>
- Wavestone. (2024). Digital sobriety: How to reduce the environmental impact of digital usage?
<https://www.wavestone.com/en/insight/digital-sobriety-how-to-reduce-the-environmental-impact-of-digital-usage/>
- Worldline. (2025). Universal registration document 2024.
https://investors.worldline.com/content/dam/investors-worldline-com/assets/documents/universal-registration-document/worldline_urd_2024.pdf

¡Muchas gracias!

Nos vemos en la próxima lección

