

# ITINERARIO PARA LA EMPLEABILIDAD 2

CASO PRACTICO UD3



ALUMNO CESUR 25/26

Alejandro Muñoz de la Sierra

PROFESOR

Paloma Ales Hermosa

# INTRODUCCION

Las empresas tecnológicas enfrentan una paradoja. Construimos el futuro digital, pero ignoramos la gran huella ecológica de nuestra actividad. El sector del software solía desatender la sostenibilidad. Creíamos que nuestro impacto era nulo por la falta de chimeneas o fábricas pesadas. La realidad es distinta. Los grandes centros de datos, la infraestructura en la nube y la rotación de equipos consumen mucha energía global.

Nuestra empresa desarrolla aplicaciones multiplataforma competitivas. No podemos operar con un modelo tradicional centrado solo en la velocidad de entrega y la reducción de costes inmediatos. La dirección entiende que integrar los Objetivos de Desarrollo Sostenible (ODS) de la Agenda 2030 es una ventaja competitiva. Los grandes clientes exigen proveedores que cumplan con criterios ESG en sus licitaciones. El talento joven prefiere organizaciones con un propósito claro más allá de las ganancias.

Hemos decidido replantear nuestro modelo de negocio. Este plan estratégico detalla la hoja de ruta para implementar una cultura de innovación sostenible. Evitamos medidas superficiales como poner papeleras de reciclaje. Nuestro objetivo es transformar la actividad central. Esto incluye la escritura de código, la gestión de infraestructura en la nube y la renovación de hardware. Queremos demostrar que la eficiencia de recursos y el impacto social positivo impulsan la rentabilidad futura en el sector IT.

# ANÁLISIS DE ÁREAS DE IMPACTO Y DESAFÍOS

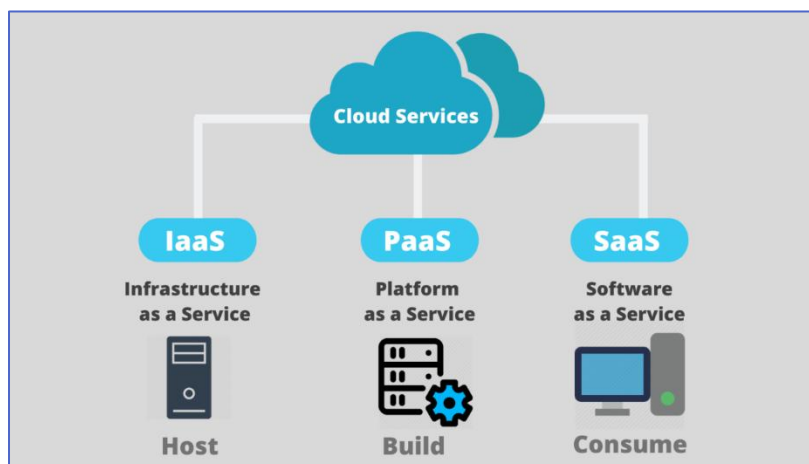
Una auditoría interna identificó tres áreas operativas clave. La adopción de prácticas sostenibles en estas áreas generará impacto positivo y equilibrará el crecimiento con la responsabilidad ambiental:

## **A. Área de Infraestructura Cloud y Sistemas (Innovación en Procesos)**

**Justificación:** Nuestros servidores operan las 24 horas. Procesan miles de peticiones de madrugada cuando nadie los usa. Migraremos la infraestructura a proveedores de nube con energía renovable, como AWS, Azure o Google Cloud. Esto reducirá nuestra huella de carbono de alcance 3. Implementaremos arquitecturas Serverless y scripts de automatización. Estos apagarán los entornos de prueba durante las noches y fines de semana. Esta medida protege el medio ambiente y ahorra dinero en la factura de Amazon.

**Desafíos:** El principal obstáculo es el riesgo técnico. Las migraciones pueden causar cortes de servicio y afectar a los clientes. El equipo de sistemas podría resistirse al cambio de arquitectura.

**Solución:** Planeamos una migración progresiva en madrugadas. Aislaremos los servicios en contenedores Docker. Tendremos un plan de "rollback" detallado por si ocurren fallos.



## **B. Área de Desarrollo de Software (Innovación en Producto y Servicio)**

**Justificación:** Adoptaremos la filosofía de Green Coding. Verde en todos nuestros squads. Un código mal escrito tiene bucles infinitos, fugas de memoria o consultas lentas. Esto fuerza a los procesadores del servidor a trabajar al 100%. Entonces consumen mucha más electricidad para refrigerarse. Enseñaremos a nuestros desarrolladores a escribir algoritmos limpios y rápidos. Así integramos la sostenibilidad desde el inicio. Un software ligero gasta menos energía. También carga más rápido en el móvil del cliente y mejora la retención.

**Desafíos:** El reto es cultural. Los programadores priorizan cerrar tareas para cumplir plazos muy ajustados. Dejan las mejoras técnicas para un futuro que nunca llega.

**Solución:** Incentivaremos la iniciativa propia para combatir esto. Premiaremos a los equipos que reduzcan el peso y el tiempo de respuesta de las aplicaciones. Integraremos herramientas automáticas de métricas de energía en las revisiones semanales de código.

### Core Green Coding Principles

- 1. Optimize Algorithms & Data Structures**

Every line of code uses real resources. Efficient algorithms, loops, and lower background processing lower energy usage. Poor choices can use 10-70x more energy!
- 2. Reduce Memory & CPU Usage**

Measurement is key. Profile CPU, memory, execution time, and energy per transaction to identify bottlenecks. Optimize continuously.
- 3. Avoid Waste by Design**

Eliminate unnecessary API calls, polls, unused dependencies, and over-engineered abstractions to lead to easier maintenance at the design stage.
- 4. Eliminate Idle & Background Processes**

Simplicity reduces complexity. Fewer moving parts mean less resources consumed, leading to easier and faster execution.
- 5. Choose Efficient Runtimes & Frameworks**

Languages like Rust/C++ are energy-efficient for heavy tasks. Modern Next.js outperforms older setups.  
Example: Flutter + Node.js reduced cloud costs by ~30%.

## C. Área de Compras y Hardware (Cambio Organizacional y Economía Circular)

**Justificación:** La industria tecnológica genera toneladas de basura electrónica muy contaminante cada año. Aplicaremos un modelo de economía circular. No renovaremos los portátiles cada dos o tres años por sistema. Maximizaremos su vida útil. Ampliaremos componentes clave como la memoria RAM o sustituiremos baterías. Un equipo puede dejar de servir para compilar código pesado. No lo tiraremos. Lo reacondicionaremos con Linux. Luego lo donaremos a escuelas locales u ONG para reducir la brecha digital.

**Desafíos:** Algunos empleados se quejarán. Los perfiles Senior suelen considerar un derecho tener el último modelo de Mac o PC. Lo ven como un símbolo de estatus.

**Solución:** Haremos campañas de comunicación interna directas. Mostraremos el impacto de la extracción de metales en países en desarrollo. Demostraremos con datos que los equipos actualizados sirven para programar a diario.





# METODOLOGÍA DE INNOVACIÓN Y PLAN DE VALIDACIÓN

Queremos implementar el Green Coding y reducir el consumo de nuestros servidores. Hemos descartado las metodologías tradicionales en cascada. Las planificaciones rígidas a un año suelen fallar en tecnología. Necesitamos un enfoque ágil y científico. Por eso usaremos Lean Startup. Así experimentaremos y aprenderemos rápido. No gastaremos dinero en ideas que no funcionen.

## Desarrollo e implementación mediante Lean Startup:

No paralizaremos la empresa para escribir un manual de sostenibilidad de 200 páginas. Comenzaremos con un Producto Mínimo Viable (MVP).

Nuestro MVP será un pequeño "Dashboard de Sostenibilidad" interno. Este panel se conectará a una sola aplicación principal. Medirá el consumo de CPU y memoria en tiempo real. También calculará el CO2 generado por cada petición.



## Proceso de Validación antes de la implementación total:

**Fase Piloto (Construir):** Un equipo voluntario usará este panel a diario durante cuatro semanas. Limpiarán y mejorarán su código según los avisos del sistema.

**Medición y Feedback (Medir):** Monitorizaremos los resultados reales en los servidores. ¿Ha bajado el consumo de CPU? ¿La aplicación sigue siendo estable? y rápida? Sobre todo, recogeremos la opinión cualitativa del equipo. Queremos saber si mirar el panel retrasa su ritmo de entrega o si resulta un reto motivador.

**Iteración o Pivote (Aprender):** Si los datos tras el mes de prueba muestran que la app es más fluida y el servidor de Amazon consume un 15% menos, habremos validado la idea. Entonces documentaremos el proceso y lo extenderemos al resto de la empresa. Si el piloto falla y retrasa las entregas un 50%, habremos gastado poco tiempo. Buscaremos otra solución sin alterar a toda la organización.



# OBJETIVOS ODS, SEGUIMIENTO Y AJUSTES

Este plan no debe quedar olvidado en un archivo. Hemos definido tres objetivos para el próximo ejercicio. Son medibles, alcanzables y están unidos a la Agenda 2030:

**Objetivo 1 (ODS 9: Industria, Innovación e Infraestructura):** Reducir el consumo energético y la factura de nuestra infraestructura cloud un 25% para el cierre fiscal. Refactorizaremos código redundante y apagaremos automáticamente los entornos no productivos fuera del horario laboral.

**Objetivo 2 (ODS 12: Producción y Consumo Responsables):** Reutilizar o reacondicionar el 80% del hardware. Bajaremos el presupuesto para equipos nuevos. Garantizaremos con certificados que los equipos descartados se donen a ONGs o se reciclen correctamente.

**Objetivo 3 (ODS 4: Educación de Calidad y ODS 8: Trabajo Decente):** Formar a toda la plantilla técnica en arquitectura de software sostenible. Realizaremos talleres internos cada dos meses en horario laboral para apoyar su crecimiento profesional.



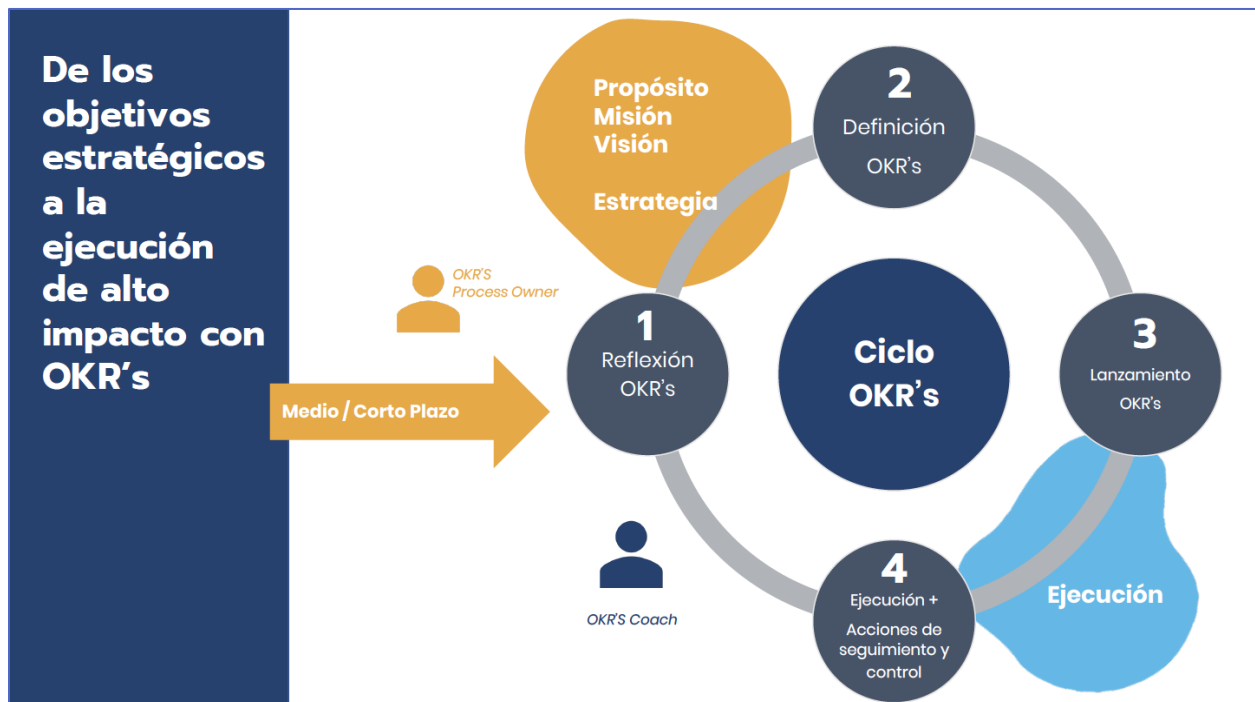


## Método de seguimiento y ajuste estratégico:

**Un directivo no llevará el control en solitario.** Fijaremos un sistema de OKRs y lo revisaremos cada mes en un comité con Recursos Humanos, líderes técnicos y Dirección.

**Usaremos el trabajo colaborativo.** Mostraremos los datos de ahorro de cada equipo en las pantallas de la oficina para mantener la transparencia y motivar al personal.

Si las **revisiones trimestrales** muestran fallos graves, no penalizaremos a los empleados. Asumiremos que el sistema falla e identificaremos el bloqueo. Ajustaremos la estrategia con inversiones en herramientas de Inteligencia Artificial para auditar código. También podríamos reasignar recursos a los equipos saturados.



## CONCLUSIONES

El cambio hacia un modelo sostenible presenta dificultades operativas. Los beneficios a largo plazo compensan las molestias iniciales. Este plan estratégico confirma que la sostenibilidad en el desarrollo es más que ecología. Es una estrategia de supervivencia, eficiencia y rentabilidad.

Aplicamos métodos reales como la auditoría de Green Coding y la economía circular. No solo mitigamos nuestro. Controlamos el impacto medioambiental y cumplimos con los Objetivos de Desarrollo Sostenible (ODS). Reducimos mucho los costes operativos mensuales en servidores de Amazon y Azure. También gastamos menos en portátiles. Esto aumenta nuestro margen de beneficio operativo (EBITDA). Podemos y debemos reinvertir este capital para mejorar las condiciones laborales del equipo. Así retenemos el talento y financiamos nuevos proyectos de I+D.

Las metodologías ágiles como Lean Startup nos enseñaron una lección importante. Los cambios no deben asustar a la plantilla. El trabajo debe ser iterativo y probado con datos reales. Involucramos a los empleados desde la base. Apoyamos sus iniciativas y escuchamos sus opiniones. La cadena de producción sigue activa. El trabajo en equipo y la transparencia de datos vencen la resistencia al cambio. Esta resistencia ocurre en cualquier transición tecnológica.

Este plan demuestra que la rentabilidad económica y la responsabilidad social corporativa son compatibles. Ambos conceptos se refuerzan mutuamente. Los desarrolladores y profesionales técnicos diseñamos el futuro digital. Mejoramos una consulta a una base de datos. Bajamos el peso de una aplicación móvil un par de megabytes. Estos cambios parecen pequeños. La eficiencia se convierte en la norma de la empresa y el impacto global es inmenso. Lideraremos este cambio desde el código. Seremos líderes en tecnología y compromiso social en el mercado.

## REFERENCIAS

<https://www.un.org/sustainabledevelopment/es/>

<https://www.un.org/sustainabledevelopment/es/economic-growth/>

<https://www.un.org/sustainabledevelopment/es/infrastructure/>

<https://www.un.org/sustainabledevelopment/es/sustainable-consumption-production/>

<https://greensoftware.foundation/>

<https://www.bbva.com/es/innovacion/que-es-el-green-software-y-cuales-son-sus-beneficios/>

<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/green-coding/>

<https://labrender.com/green-coding-como-reducir-la-huella-de-carbono-en-el-desarrollo-de-software/>

**Lean Startup | Qué Es Y Cómo Funciona El Método Lean Startup**

<https://www.youtube.com/watch?v=xhXyGAkatuM>

**Proceso para aplicar Lean Startup - Gestión del emprendimiento - Udla en línea**

<https://www.youtube.com/watch?v=ryqXm14oFj0>