



Sesión 4

Juan Díaz de la Vega Pérez A00575311

Patricio Hinojosa Legaspi A00575222

Fabricio Padilla

Enrique Camarena

Emiliano Ramírez

Resumen del problema y criterios de éxito (de la Sesión 2)

La contaminación marina por plásticos carece de un sistema accesible y colaborativo que permita a pescadores, turistas y ciudadanos reportar y detectar basura, lo que limita la protección de especies y ecosistemas marinos.

ODS seleccionado: ODS 14 — Vida submarina.

Población objetivo: pescadores, turistas y organizaciones ambientales.

Criterios de éxito del prototipo:

1. Detectar basura en imágenes con al menos 80% de precisión.
2. Interfaz sencilla para usuarios sin experiencia tecnológica.
3. Buscador de reportes por palabra clave.
4. Sistema de inicio/cierre de sesión con reportes guardados.
5. Gamificación: puntos, logros, tabla de posiciones, recompensas simbólicas.

Lluvia de ideas de soluciones

- Aplicación móvil para que los usuarios suban fotos de basura detectada en playas o mares.
- Plataforma web colaborativa para mapear puntos críticos de contaminación marina.
- Integración con IA para detectar automáticamente basura en imágenes subidas.
- Sistema de recompensas (puntos, medallas digitales, reconocimiento en redes sociales).
- Chatbot educativo sobre contaminación marina y reciclaje.

- API abierta para que organizaciones ambientales accedan a datos de reportes.
- Módulo de visualización con mapas y gráficas (zonas más afectadas, tipos de desechos).
- Función de alerta a pescadores y comunidades locales cuando se detecta acumulación de basura.

Idea elegida y justificación

La aplicación móvil y web híbrida con detección de basura mediante IA y sistema de reportes colaborativos.

Justificación:

- Responde directamente al ODS 14 y al problema de la falta de sistemas accesibles de monitoreo.
- Cumple los criterios de éxito al incluir detección automática, usabilidad sencilla y gamificación.
- Impacta positivamente en pescadores, turistas y organizaciones ambientales, creando un sistema participativo y motivador.



Clasificación MoSCoW

Categoría	Funcionalidades
Must	<ul style="list-style-type: none">- Registro de usuarios (básico)- Reporte de basura/contaminación con foto y ubicación- Visualización en un mapa de los reportes- Sección educativa con información breve sobre impacto ambiental
Should	<ul style="list-style-type: none">- Ranking de usuarios más activos- Consejos personalizados para reducir plásticos- Compartir reportes en redes sociales
Could	<ul style="list-style-type: none">- Gamificación (insignias, niveles, recompensas)- Chat comunitario- Traducción a otros idiomas
Won't	<ul style="list-style-type: none">- Compatibilidad con sistemas operativos antiguos- Funcionalidades premium/de pago- Integración con drones o sensores especializados

Wireflow (Flujo de pantallas con PySimpleGUI)

1. Pantalla de bienvenida / Inicio

- **Contenido :** Título de la aplicación ("Mares Limpios"), una breve introducción sobre el ODS 14 y el problema de la basura marina.
- **Componentes :**
 - `sg.Text`: Mensaje de bienvenida.
 - `sg.Button`: "Iniciar Sesión"
 - `sg.Button`: "Registrarse"
- **Flujo :**
 - Si el usuario hace clic en "Iniciar Sesión" , aparecerá la pantalla de inicio de sesión.
 - Si el usuario hace clic en "Registrarse" , aparecerá la pantalla de registro.

2. Pantalla de registro

- **Contenido:** Formulario para que un nuevo usuario cree una cuenta.
- **Componentes:**
 - `sg.Input`: Para nombre de usuario y contraseña.
 - `sg.Button`: "Crear Cuenta".
 - `sg.Button`: "Cancelar" (Regresar a la pantalla de bienvenida).
- **Flujo:**
 - El usuario introduce sus datos. El módulo `core/` valida la información.
 - Si el registro es exitoso:
 - Los datos se guardan en el módulo `data/`.
 - Se redirige al usuario a la pantalla principal.
 - Si el registro falla (ej. usuario ya existe):
 - Se muestra un `sg.Popup` de error.

3. Pantalla principal / Dashboard

- **Contenido:** Muestra un resumen de la actividad del usuario y el impacto total.
- **Componentes:**
 - `sg.Text`: "Bienvenido, [nombre de usuario]".
 - `sg.Text`: Resumen de recolecciones realizadas.
 - `sg.Button`: "Registrar Recolección".
 - `sg.Button`: "Ver Estadísticas".
 - `sg.Button`: "Cerrar Sesión".
- **Flujo:**
 - El módulo `data/` carga las recolecciones del usuario.
 - El módulo `viz/` puede generar y mostrar un gráfico simple aquí si se desea.
 - Si el usuario hace clic en "Registrar Recolección", va a la pantalla de registro de basura.
 - Si el usuario hace clic en "Ver Estadísticas", va a la pantalla de estadísticas detalladas.
 - Si el usuario hace clic en "Cerrar Sesión", regresa a la pantalla de bienvenida.

4. Pantalla de registro de recolección

- **Contenido:** Formulario para registrar los detalles de una nueva recolección de basura.
- **Componentes:**
 - `sg.Input`: Para el tipo y la cantidad de basura recolectada.
 - `sg.Button`: "Guardar Recolección".
 - `sg.Button`: "Cancelar" (Regresar al dashboard).
- **Flujo:**

- El usuario introduce la información.
- El módulo `core/` procesa la información y calcula el impacto (por ejemplo, puntos o equivalencia en plástico).
- El módulo `data/` guarda el nuevo registro.
- Se redirige al usuario a la pantalla principal.

5. Pantalla de estadísticas

- **Contenido:** Visualización detallada del progreso del usuario y el impacto global.
- **Componentes:**
 - `sg.Canvas`: Para incrustar gráficos de `matplotlib` generados por el módulo `viz/`.
 - `sg.Text`: Métricas clave (por ejemplo, kilos de plástico recolectados).
 - `sg.Button`: "Volver" (Regresar al dashboard).
- **Flujo:**
 - El módulo `viz/` solicita datos al módulo `data/` y al `core/` para generar gráficos.
 - El usuario ve los gráficos y la información.
 - Al hacer clic en "Volver", regresa a la pantalla principal.

