**FORMATO N° 04**

**INFORME TÉCNICO DE PRÁCTICAS PRE-PROFESIONALES**

**QUE PRESENTA EL ESTUDIANTE[[1]](#footnote-1)**

1. **PORTADA**



**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN**

**CARRERA DE INGENIERIA EN SOFTWARE**

**INFORME DE:**

****

**NOMBRE DE LA EMPRESA/ INSTITUCIÓN/ COMUNIDAD DONDE REALIZÓ LA**

**PRÁCTICA PRE PROFESIONAL**

**NOMBRES Y APELLIDOS DEL ESTUDIANTE: MATEO JAVIER CONDOR SOSA**

**NOMBRES Y APELLIDOS DEL TUTOR ACADÉMICO: JENNY ALEXANDRA RUIZ ROBALINO**

**CALIFICACIÓN DEL INFORME**

**FIRMA DE LA TUTORA ACADÉMICA FIRMA DEL ESTUDIANTE**

**FIRMA DEL TUTOR EMPRESARIAL**

**Sangolquí, 15 de julio de 2025**

1. **INTRODUCCIÓN**

En los últimos años, la contaminación de los ríos ha representado una creciente preocupación ambiental en Ecuador. La acumulación de residuos sólidos en cuerpos de agua no solo afecta gravemente a los ecosistemas acuáticos, sino que también compromete la salud pública y el desarrollo sostenible de las comunidades cercanas. Frente a esta problemática, surge la necesidad de implementar soluciones tecnológicas que permitan visualizar, analizar y gestionar de manera efectiva la información recolectada durante las jornadas de limpieza y clasificación de residuos.

En este marco, realicé mis prácticas preprofesionales desde el 7 de abril hasta el 4 de julio de 2025, participando activamente en el análisis, diseño, maquetación y desarrollo de un dashboard estadístico enfocado en la visualización de datos sobre desechos recolectados por ríos y clasificados mediante inteligencia artificial. Este trabajo se desarrolló en el contexto de un proyecto ambiental con fines investigativos, donde la recolección de datos es parte fundamental para evaluar el impacto de la contaminación y proponer acciones correctivas.

La institución donde se desarrolló la práctica promueve proyectos tecnológicos con enfoque social y ambiental, y cuenta con un equipo interdisciplinario de trabajo. Durante mi permanencia, pude integrarme a un entorno colaborativo en el que comprendí tanto los procesos manuales de recolección de datos como los procesos automatizados de clasificación por IA. Asimismo, se detectaron oportunidades de mejora en la gestión y presentación de datos, lo que impulsó el desarrollo del dashboard como una solución pertinente.

Estas actividades se enmarcan plenamente dentro del perfil de egreso de la carrera de Ingeniería en Software, al implicar la aplicación de conocimientos en desarrollo de software, diseño de interfaces gráficas, análisis de datos y trabajo colaborativo. Además, fortalecieron mis capacidades técnicas y analíticas, permitiéndome contribuir con una herramienta funcional al proceso de monitoreo ambiental. El objetivo principal de este trabajo fue desarrollar una solución que permitiera transformar datos complejos en información útil para la toma de decisiones, promoviendo así un uso más eficiente de la información recolectada y contribuyendo a la sostenibilidad ambiental.

1. **DESARROLLO**

Durante el periodo de pasantías, participé activamente en el departamento de tecnología en el cual llevé a cabo mis actividades enfocadas en el análisis, diseño, maquetación y desarrollo de un dashboard estadístico para la visualización de residuos recolectados en cuerpos de agua y clasificados mediante inteligencia artificial. Este proceso se dividió en tres fases claramente estructuradas, que abarcaron desde el estudio del sistema de clasificación hasta la implementación funcional del dashboard. A continuación, se describen las actividades realizadas, herramientas utilizadas, metodología aplicada, así como los logros y desafíos enfrentados en cada fase.

* 1. **Fase 1: Análisis de la Clasificación**

Durante la primera fase de mi práctica preprofesional, me desempeñé en el área de Análisis y Clasificación de Datos, formando parte del equipo encargado del tratamiento inicial de los resultados generados por un modelo de inteligencia artificial orientado a la clasificación automatizada de residuos recolectados en ríos. Esta etapa fue fundamental para comprender el funcionamiento del sistema, definir los requerimientos de visualización y establecer las bases para el diseño del dashboard estadístico.

**Metodología y Técnicas Aplicadas**

Se empleó una **metodología iterativa**, con reuniones periódicas de revisión y validación. Se utilizó el enfoque **bottom-up**, partiendo del análisis individual de los resultados del modelo para luego escalar hacia una propuesta estructural jerárquica de los residuos (superclase, clase, subclase). Las técnicas aplicadas incluyeron:

* Lectura y análisis de la matriz de confusión.
* Identificación de patrones semánticos y conflictos en las etiquetas.
* Clasificación jerárquica de residuos.
* Propuestas de reorganización y redefinición de categorías.
* Validaciones cruzadas con el equipo técnico y de diseño.

**Actividades Realizadas**

1. **Inducción y comprensión del proyecto:** Asistí a una reunión de bienvenida donde se presentaron los lineamientos institucionales y se explicó a detalle el objetivo del sistema.
2. **Exploración del sistema actual:** Se analizaron documentos técnicos, esquemas de arquitectura, y se visualizaron salidas reales del modelo con etiquetas acompañadas de su nivel de confianza.
3. **Definición de objetivos del dashboard:** Definí los indicadores clave que deberían visualizarse, identificando necesidades del usuario final y generando una primera guía de gráficos requeridos.
4. **Análisis de resultados del modelo IA:** Se estudió la matriz de confusión para detectar aciertos y errores en la clasificación por categoría como se puede observar en la Figura 1.
5. **Agrupamiento jerárquico de residuos:** Se propuso una estructura jerárquica de tres niveles (superclase → clase → subclase) como se muestra en la Figura 2.
6. **Definición de nomenclatura y visualización:** Se asignaron códigos de colores, abreviaciones y nombres representativos a cada superclase que se pueden observar en la Figura 2.

**Calendario

El contenido generado por IA puede ser incorrecto.**

Figura Fragmento de la matriz de confusión analizada con observaciones clave

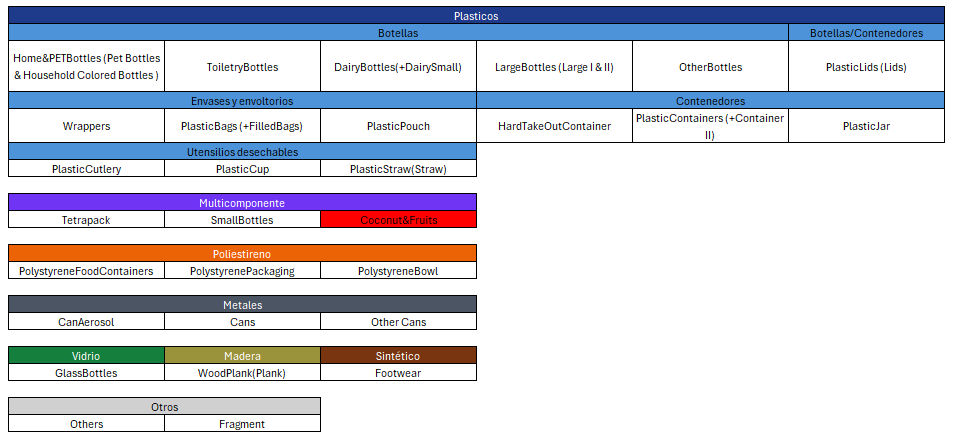
****

Figura Estructura jerárquica propuesta para las categorías de residuos

* 1. **Fase 2: Diseño y Maquetación del Dashboard**

Una vez validada la estructura de clasificación jerárquica de los residuos, se procedió al diseño visual y funcional del dashboard estadístico. Esta fase se desarrolló enfocándose en la creación de una experiencia visual clara, coherente y funcional para los usuarios finales del sistema.

**Metodología y Técnicas Aplicadas**

Se empleó la metodología SCRUM con un enfoque de diseño centrado en el usuario, validando progresivamente las propuestas de interfaz con el tutor y usuarios internos. Las herramientas utilizadas fueron Figma para diseño de interfaces y Seaborn / Matplotlib para la generación de gráficos. Las principales técnicas incluyeron:

* Prototipado rápido en baja y alta fidelidad.
* Documentación visual con versiones antes/después.
* Diseño adaptativo basado en jerarquía de la información y principios de visualización efectiva.

**Actividades Realizadas**

1. **Inspiración y buenas prácticas:** Se revisaron ejemplos de dashboards y bibliografía sobre visualización efectiva. Esto sirvió como referencia para el diseño de estructuras intuitivas y minimalistas.
2. **Bocetado en Figma:** Como se pueden observar en la Figura 3 se realizaron los primeros prototipos que incluían:

* Vista general con resumen de residuos.
* Vistas por categoría y filtros interactivos.
* Detalles por subclase.

1. **Diseño visual y navegación:** Se ajustaron paletas de colores, tipografías, márgenes y componentes visuales (filtros, íconos, contenedores) buscando consistencia visual y facilidad de navegación.
2. **Reuniones de validación visual:** Se presentaron las propuestas al tutor técnico, quien aportó retroalimentación útil. Como resultado, se reorganizaron secciones y se implementaron mejoras visuales sugeridas como se observa en la Figura 4.
3. **Documentación de mejoras:** Se capturaron versiones "antes y después" para mostrar la evolución del diseño, incluyendo cambios en la estructura y flujo de navegación.
4. **Diseño de gráficos de prueba:** Se generaron visualizaciones con datos ficticios para probar la integración visual de los gráficos en las vistas del dashboard.
5. **Implementación de funciones gráficas:**

* Se crearon funciones parametrizadas en Python para generar gráficos como barras, treemaps, sunbursts y gauges.
* Se probaron estilos visuales con Seaborn y se definieron márgenes, colores y escalas.

1. **Validación visual de gráficos integrados:** Se probaron las visualizaciones exportadas dentro del prototipo de Figma, evaluando la coherencia estética entre el diseño de interfaz y los gráficos reales.
2. **Ajustes finales y visualización del diseño ensamblado:** Se ensamblaron las diferentes secciones del dashboard, con los gráficos ya adaptados a su contexto visual final. Esto permitió validar que cada componente gráfico estaba bien posicionado, legible y acorde con la intención comunicativa.

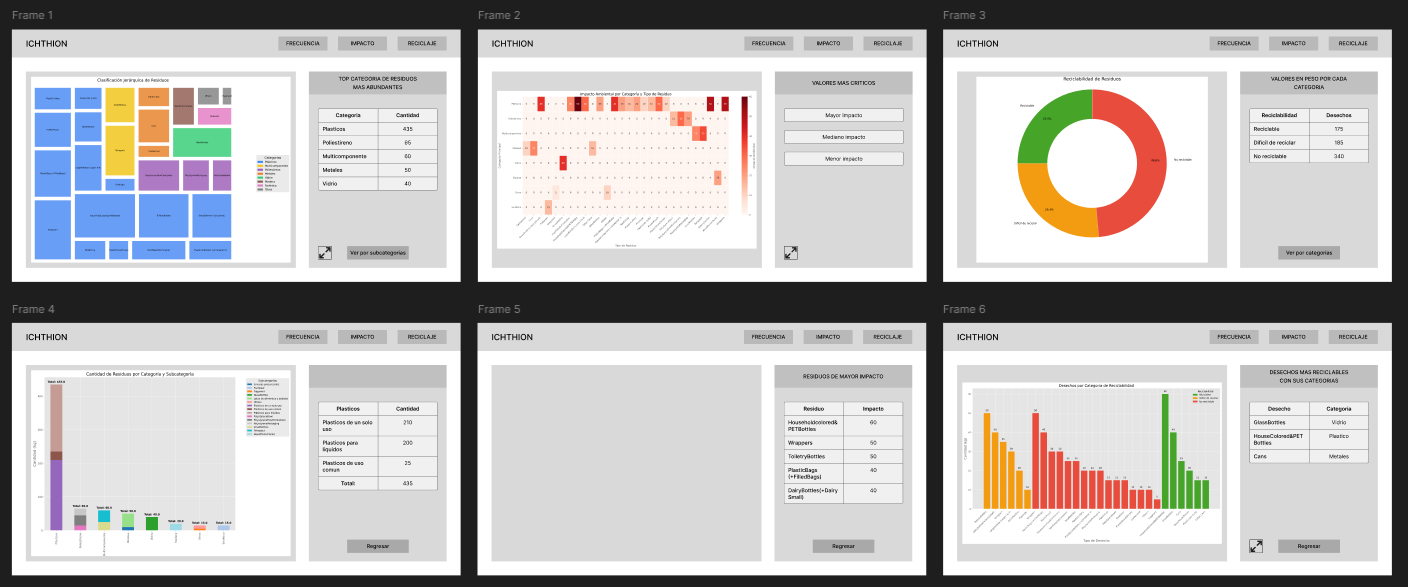
****

Figura Primer prototipo previo cambios sugeridos por el tutor

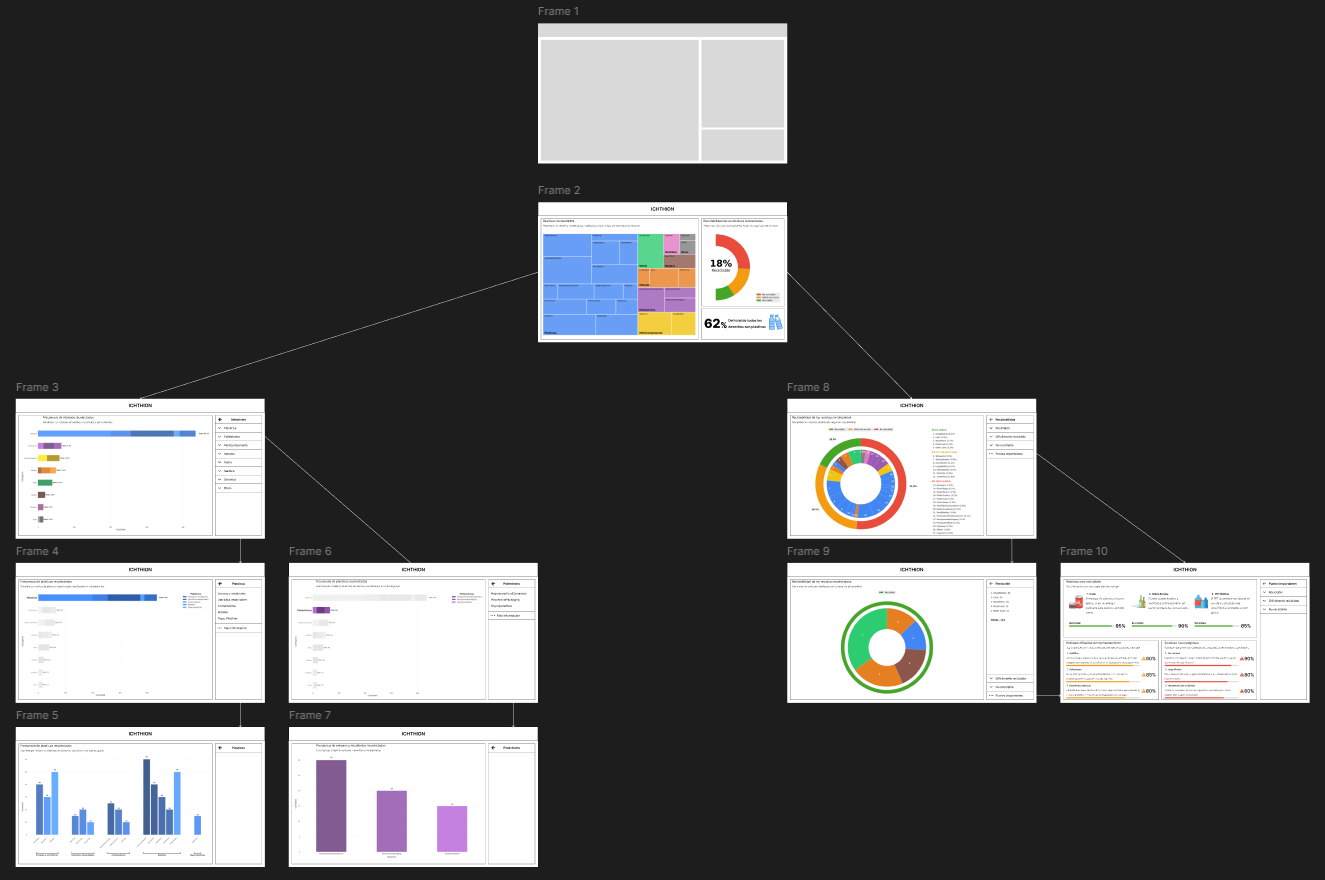
****

Figura Segundo prototipo después de aplicar cambios

* 1. **Fase 3: Desarrollo del Dashboard**

La tercera fase se centró en la implementación técnica del prototipo diseñado, utilizando herramientas de desarrollo multiplataforma. Esta etapa se llevó a cabo en el área de **Desarrollo de Software**, específicamente en el desarrollo de interfaces gráficas y su integración con módulos de visualización de datos. El objetivo fue construir una aplicación funcional que permita al usuario explorar datos de clasificación de residuos mediante gráficos interactivos.

**Metodología y Técnicas Aplicadas**

Se continuó con una metodología modular, permitiendo integrar y probar progresivamente las funcionalidades del dashboard. Las principales técnicas utilizadas incluyeron:

* Contenerización del entorno usando Docker.
* Desarrollo de interfaces gráficas con Tkinter.
* Integración de visualizaciones con Matplotlib mediante FigureCanvasTkAgg.
* Programación estructurada y modular.
* Validación de eventos e interacciones.

**Actividades Realizadas**

1. **Configuración del entorno de desarrollo**

* Instalación de **Docker Desktop** y creación de un contenedor con imagen base de Python y librerías requeridas como se muestra en la Figura 5.
* Configuración de **Xming/XLaunch** para ejecución gráfica desde contenedor.
* Verificación de ejecución correcta de scripts y visualización de interfaces desde VSCode dentro del contenedor.

1. **Diseño inicial de la interfaz con Tkinter**

* Se creó la ventana principal con estructura modular: títulos, separadores, placeholders para gráficos como se observa en la Figura 7.
* Separación de componentes en archivos independientes para facilitar mantenimiento como se puede ver en la Figura 6.
* Programación de botones y navegación básica entre vistas.

1. **Integración de gráficos**

* Uso de FigureCanvasTkAgg para mostrar gráficos generados por Matplotlib directamente en la interfaz.
* Configuración de alineación automática, redimensionamiento, y actualización dinámica de contenido.

1. **Diseño de navegación dinámica**

* Incorporación de filtros (superclases, clases) que actualizan los gráficos mostrados en pantalla.
* Validación de comportamiento adecuado al cambiar de categoría o aplicar filtros consecutivos.

1. **Optimización del flujo de visualización**

* Se desarrolló lógica para cerrar gráficos anteriores, liberar memoria y evitar superposición de visualizaciones.
* Se añadieron mensajes de placeholder cuando no hay datos disponibles para mostrar.

1. **Incorporación de tipos de gráficos**

* Gráfico treemap y gauge para resumen general como se muestra en la Figura 7.
* Gráfico de barras y sunburst para subclases como se muestra en la Figura 8 y Figura 9.

1. **Ajustes de visualización y estructura**

* Corrección de títulos, leyendas, escalas y formato de datos.
* Reorganización del código y mejora de la paleta de colores, rotación de etiquetas y espaciado interno.

1. **Pruebas funcionales del sistema**

* Navegación completa por las distintas vistas como se puede observar en las figuras 7, 8 y 9.
* Validación de todos los componentes: filtros, botones, gráficos y etiquetas.

1. **Reuniones de revisión y mejora**

* Se realizaron varias reuniones para evaluar el estado funcional del dashboard.
* Se identificaron mejoras visuales y de rendimiento.
* Se reemplazaron funciones antiguas por versiones modulares más eficientes y validadas.

Texto

El contenido generado por IA puede ser incorrecto.

Figura Archivos de configuración del entorno de desarrollo

Texto

El contenido generado por IA puede ser incorrecto.

Figura Estructura raíz de carpetas del proyecto

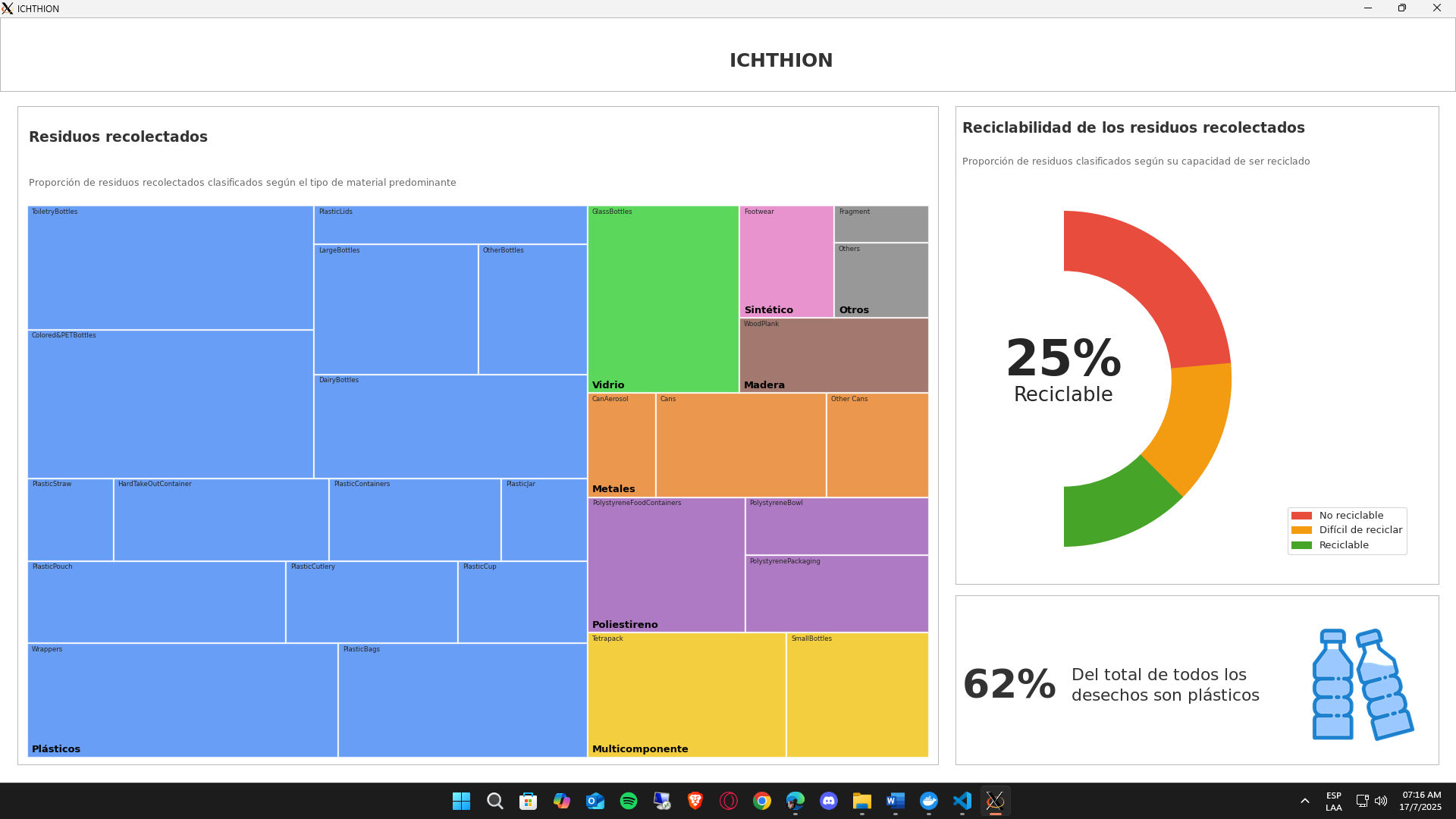
****

Figura Interfaz principal del dashboard construida con Tkinter

**Gráfico

El contenido generado por IA puede ser incorrecto.**

Figura Navegación a residuos según filtros aplicados

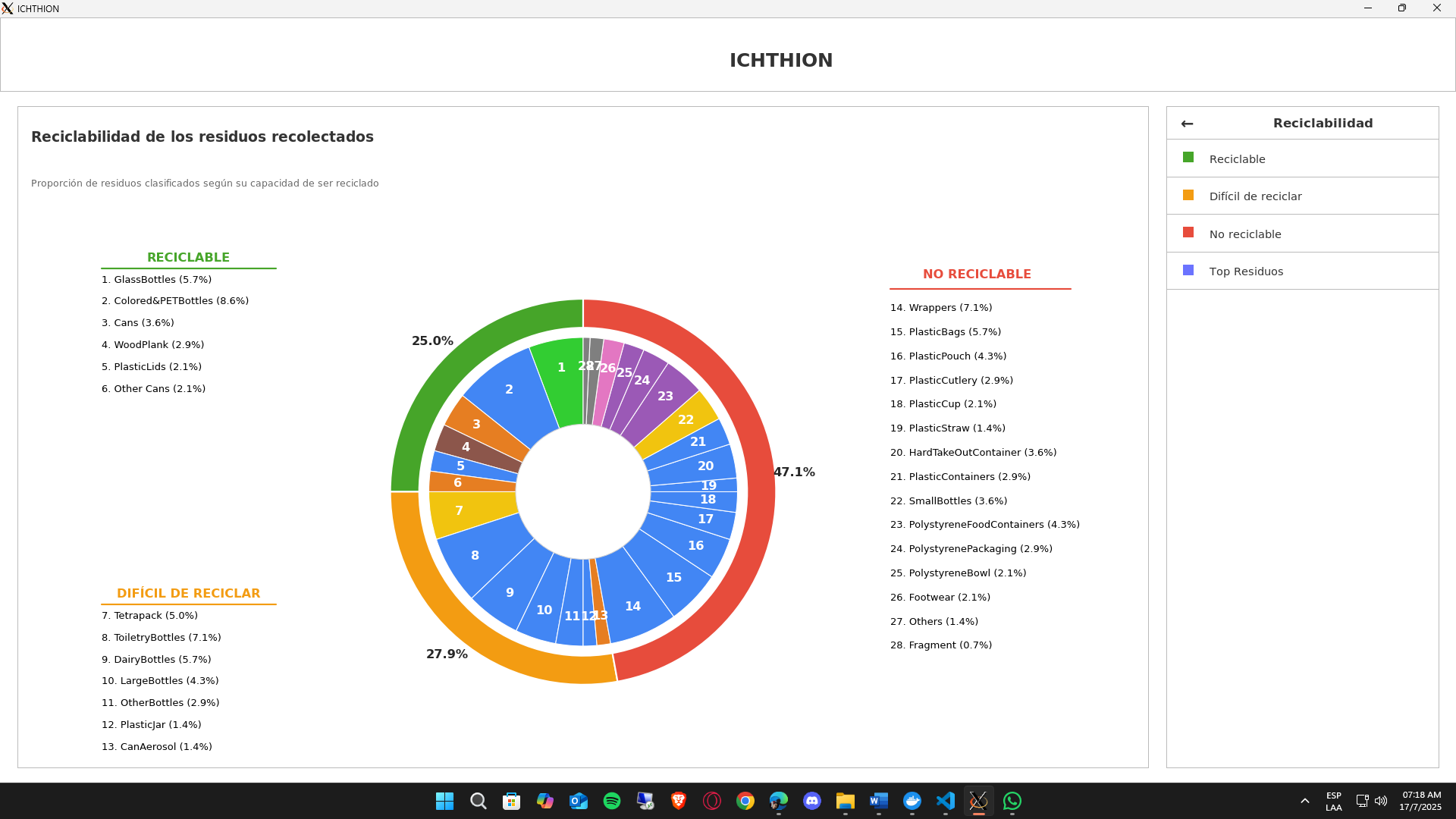
****

Figura Navegación a reciclabilidad según filtros aplicados

* 1. **Recursos Utilizados**
* Documentación del sistema de IA y matrices de resultados.
* Figma (prototipado UI/UX)
* Docker Desktop
* Python (Tkinter, Matplotlib, Seaborn)
* Xming y XLaunch (visualización remota).
* VSCode con conexión a contenedor Docker.
  1. **Limitaciones Encontradas**
* Dificultad en validar algunas subclases debido a escasez de ejemplos o ambigüedad en la clasificación original.
* Las visualizaciones con gran densidad de datos requerían ajustes finos en márgenes y escalas para evitar saturación visual.
* Las paletas de colores iniciales no siempre combinaban con el diseño, requiriendo iteraciones para mejorar la estética.
* Algunos gráficos no se integraban correctamente con las proporciones de la maqueta, requiriendo redimensionamiento manual.
* Problemas iniciales con la visualización de interfaces Tkinter desde contenedor Docker (resueltos configurando correctamente el display).
* Necesidad de ajustar manualmente ciertos márgenes y posiciones en el layout de Tkinter.
  1. **Éxitos Alcanzados**
* Consolidación de una estructura jerárquica clara y validada, con nomenclatura homogénea.
* Aprobación de la propuesta organizacional del sistema de categorías por parte del tutor empresarial.
* Se logró una maqueta completamente funcional y visualmente coherente, lista para ser desarrollada.
* Las funciones gráficas quedaron parametrizadas y alineadas con el estilo visual del dashboard.
* Validación del flujo de navegación y estructura por parte del tutor empresarial.
* Se completó un prototipo funcional del dashboard, navegable y con integración de filtros y gráficos.
* Se validó el flujo de navegación y la actualización dinámica de contenido.
* El sistema está listo para conectarse con un backend o alimentarse con datos reales.

1. **CONCLUSIONES**

Durante el desarrollo de mis prácticas preprofesionales, se cumplieron satisfactoriamente los objetivos planteados, consolidando un aprendizaje tanto a nivel técnico como profesional. La experiencia me permitió participar activamente en todas las fases del desarrollo de un sistema real, desde el análisis de datos clasificados por inteligencia artificial hasta la implementación funcional de un dashboard estadístico.

El análisis exhaustivo de los datos generados por el sistema de inteligencia artificial permitió mejorar la precisión de la clasificación de residuos, estableciendo una estructura jerárquica clara y validada. Esta mejora facilitó la generación de información relevante y comprensible para apoyar la toma de decisiones ambientales.

Las propuestas de interfaz del dashboard fueron diseñadas con base en principios de usabilidad y validadas de forma iterativa con el director de Tecnología, lo cual garantizó una alineación efectiva con los requerimientos empresariales y una experiencia de usuario adecuada para perfiles técnicos.

Se logró desarrollar una aplicación funcional utilizando Python y Tkinter, integrando adecuadamente los datos procesados. El resultado fue una herramienta estadística interactiva, modular y visualmente coherente que permite explorar y visualizar los residuos clasificados de forma dinámica, aportando valor al monitoreo ambiental.

Finalmente, esta experiencia me permitió aplicar y ampliar mis conocimientos en desarrollo de software con enfoque ambiental y social, reafirmando mi compromiso como futuro ingeniero en soluciones tecnológicas con impacto positivo en la comunidad.

1. **RECOMENDACIONES**

Se recomienda fortalecer la calidad de los datos utilizados por el sistema de inteligencia artificial, especialmente en aquellas categorías con baja representación o con problemas de ambigüedad, incorporando mecanismos automáticos de validación y nuevas muestras etiquetadas correctamente. Esto permitirá mejorar la precisión del modelo y, por ende, la confiabilidad de los datos visualizados. Asimismo, sería beneficioso integrar el dashboard con una base de datos en tiempo real, lo que facilitaría la actualización continua de la información y la toma de decisiones basada en datos actuales. Esta integración también abriría la posibilidad de añadir nuevas funcionalidades, como filtros por fechas, reportes descargables o visualización geográfica.

Además, se sugiere fomentar una mayor colaboración entre las áreas técnicas, de diseño y operativas, mediante espacios de retroalimentación continua que permitan alinear mejor las necesidades de los usuarios con las soluciones implementadas. Para facilitar este proceso y asegurar la continuidad del proyecto, es importante contar con una documentación técnica clara, accesible y bien estructurada desde las primeras etapas de desarrollo. Finalmente, se recomienda a la universidad continuar fortaleciendo en su plan de estudios los conocimientos en diseño de interfaces gráficas y visualización de datos, ya que son competencias clave para el desarrollo de soluciones tecnológicas efectivas y centradas en el usuario.

1. **El informe será realizado y firmado por el estudiante y presentado a los tutores académico y empresarial, luego al coordinador de prácticas preprofesionales de la carrera y/o departamento.** [↑](#footnote-ref-1)