INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE

Departamento de Matemáticas y Física

Desarrollo tecnológico y generación de riqueza sustentable

PROYECTO DE APLICACIÓN PROFESIONAL (PAP)

PROGRAMA DE MODELACIÓN MATEMÁTICA PARA EL DESARROLLO DE PLANES Y MODELOS DE NEGOCIO

4J05A OPTIMIZACIÓN DE PROGRAMAS DE INVERSIÓN EN INTERMEDIARIOS FINANCIEROS

DASHBOARD DE PORTAFOLIOS DE INVERSIÓN IMPULSO PATRIMONIAL, ASESOR EN INVERSIONES INDEPENDIENTE

PRESENTAN

Ing. Financiera Marianne Trujillo Altamirano

Ing. Financiera Íñigo de Obeso Landeros

Ing. Financiera Karimme Anahi Tejeda Olmos

Ing. Financiera Alberto Marín Gómez

Ing. Financiera Daniel Sánchez López

Profesor PAP:

Sean Nicolás Vázquez; Luis Felipe Gómez Estrada

Tlaquepaque, Jalisco, julio de 2025

ÍNDICE

Contenido

REPORTE PAP 2
Presentación Institucional de los Proyectos de Aplicación Profesional 2
Resumen 2
1. Introducción 3
1.1. Objetivos 3
1.2. Justificación 3
1.3 Antecedentes 4
1.4. Contexto 4
2. Desarrollo 5
2.1. Sustento teórico y metodológico 5
2.2. Planeación y seguimiento del proyecto 6
3. Resultados del trabajo profesional 16
4. Reflexiones del alumno o alumnos sobre sus aprendizajes, las implicaciones éticas y los aportes sociales del proyecto 18
5. Conclusiones 26
6. Bibliografía 27
Anexos (en caso de ser necesarios) 28

REPORTE PAP

Presentación Institucional de los Proyectos de Aplicación Profesional

Los Proyectos de Aplicación Profesional (PAP) son una modalidad educativa del ITESO en la que el estudiante aplica sus saberes y competencias socio-profesionales para el desarrollo de un proyecto que plantea soluciones a problemas de entornos reales. Su espíritu está dirigido para que el estudiante ejerza su profesión mediante una perspectiva ética y socialmente responsable.

A través de las actividades realizadas en el PAP, se acreditan el servicio social y la opción terminal. Así, en este reporte se documentan las actividades que tuvieron lugar durante el desarrollo del proyecto, sus incidencias en el entorno, y las reflexiones y aprendizajes profesionales que el estudiante desarrolló en el transcurso de su labor.

Resumen

El objetivo del proyecto fue entregar a la empresa una aplicación local que sirviera como evolución de la versión desarrollada el semestre anterior, facilitando el acceso y uso del *Dashboard* para el rebalanceo de portafolios. La nueva aplicación fue desarrollada en *Electron,* utilizando *JavaScript* como lenguaje principal, lo cual implicó la migración del código anterior realizado en *Python*. Esta migración permitió entregar los instaladores ejecutables en formatos **.exe** (para *Windows*) y **.dmg** (para *Mac*).

Se mantuvo una interfaz sencilla e intuitiva, siguiendo la estética de la empresa, incorporando tablas y gráficos con colores diferenciados para proyectar los resultados del portafolio de una forma clara. Asimismo, se continuó con los indicadores visuales que sugieren las acciones a seguir (compra/venta) de activos, con el fin de apoyar la toma de decisiones para realizar un mejor rebalanceo de las inversiones de los clientes.

Dado que los archivos con los que se trabaja y se cargan a la aplicación son provenientes de distintos bancos y presentan formatos y extensiones diferentes, fue necesario mejorar el código anterior para permitir la lectura, limpieza y clasificación automática de los datos.

Finalmente, como segundo objetivo clave fue asegurar la escalabilidad de la aplicación, mejorando su flexibilidad para que pueda adaptarse a nuevos formatos o bancos sin requerir de modificaciones en el código. Con todo esto, se busca que la aplicación siga siendo funcional y útil para la empresa a largo plazo.

1. Introducción

1.1. Objetivos

El objetivo principal del proyecto fue migrar el código desarrollado durante el semestre anterior hacia una aplicación local más accesible y amigable para el usuario, utilizando la plataforma de *Electron*.

El segundo objetivo, fue optimizar y ampliar el código para que pudiera adaptarse a las diferentes estructuras en los archivos proporcionados por los diferentes bancos, por lo que se requería de una reestructuración en el flujo de procesamiento de datos de tal manera que, mediante una única función, la aplicación pudiera interpretar correctamente los archivos de los cinco bancos con los que se cuenta actualmente, así como adaptarse fácilmente a nuevos bancos o a cambios en las categorías de inversión de los bancos actuales.

Además, de realizar mejoras en la experiencia del usuario con el *Dashboard* interactivo con el fin de facilitar la visualización de resultados y apoyar la toma de decisiones financieras de manera más clara e informada para su portafolio.

1.2. Justificación

México cuenta con un nivel bajo de educación financiera, lo que convierte el simple acto de invertir en un gran logro para muchas personas. Cuando estás invirtiendo, realizar rebalanceos periódicos es necesario para mantener tu estrategia de inversión “saludable”, pero representa un reto incluso para profesionales del sector financiero ya que es una actividad compleja, especialmente cuando no se cuentan con herramientas necesarias que te faciliten el trabajo.

Ante esta situación, la aplicación desarrollada ofrece una solución automatizada, eficiente y sin costo, que simplifica el proceso de rebalanceo para Impulso Patrimonial Asesor en

Inversiones Independiente (en adelante Impulso Patrimonial) que se encarga de administrar las inversiones de múltiples clientes.

Esta aplicación es una solución que contribuye a Impulso Patrimonial para mejorar su eficiencia operativa, reduce los errores humanos y los recursos que destinaban anteriormente para realizar esta actividad; pero, también es un beneficio para todos sus clientes que ponen su dinero en sus manos para la gestión y administración de sus inversiones.

1.3 Antecedentes

El proyecto tuvo su origen con una versión inicial desarrollada durante el semestre anterior implementada en *Python* y usando la biblioteca de *Streamlit.* Sin embargo, esta versión contaba con una limitación importante, su uso requería clonar el repositorio del código en cada computadora y ejecutar la aplicación mediante la terminal, lo cual implicaba instalar dependencias y escribir líneas de comando. Este proceso resultaba poco práctico, especialmente considerando que el personal de la empresa no cuenta con conocimientos de programación, lo que dificultaba significativamente el acceso y uso del *Dashboard.*

Iniciando el semestre, se propuso desarrollar una aplicación local que pudiera ejecutarse fácilmente con un doble clic, eliminando cualquier barrera técnica para los usuarios. La mejor forma encontrada fue a través de *Electron,* una plataforma basada en lenguaje JavaScript que permite generar ejecutables tanto para sistemas *Windows* como para *MacOs.*

La colaboración entre el equipo se gestionó a través del uso de *GitHub*, ya que esta herramienta permitió contar con un seguimiento en tiempo real de los cambios, agregar comentarios sobre lo agregado o lo eliminado del código y saber quién hizo cada cambio y en qué momento.

1.4. Contexto

El proyecto se desarrolló en un ambiente académico-profesional, en el cual se compartían días de aprendizaje y la aplicación en el proyecto con las necesidades reales de una empresa del sector financiero que atiende a clientes reales.

Impulso Patrimonial, la empresa que nos permitió trabajar con ellos, compartió sus requerimientos específicos y participó en el proceso de revisión, brindando retroalimentación durante una presentación que se llevó a cabo con un representante al que se le mostraron los resultados.

Esto nos permitió como equipo el poder alinear el proyecto con las condiciones reales del sector financiero, asegurando que la solución que presentamos tuviera una aplicación real y un impacto social genuino, más allá de solo ser un proyecto escolar que será olvidado terminando el semestre.

2. Desarrollo

2.1. Sustento teórico y metodológico

El presente proyecto se enmarca en el ámbito de las finanzas, programación en *JavaScript*, *CSS*, *HTML*, visualización de datos y automatización de procesos. Su objetivo principal es facilitar el rebalanceo de portafolios financieros mediante una aplicación local, interactiva y personalizada a las necesidades de la empresa.

El rebalanceo de portafolios consiste en ajustar periódicamente la asignación de los activos de un portafolio de acuerdo con el perfil y los objetivos que se planteó el inversionista en un principio, ya que el mercado de valores es muy volátil.

Esta práctica es esencial para mantener el riesgo de un portafolio en los niveles que el inversionista desea, pero es una práctica compleja, sobre todo cuando se manejan múltiples clientes y cuando los datos provienen de diferentes lugares y de forma no estandarizada.

En cuanto a la metodología, el proyecto se enfocó en la reimplementación del sistema en un nuevo entorno de programación, utilizando *JavaScript*, *CSS* y *HTML*. Para la creación de la aplicación local se empleó *Electron* y se reestructuró el código para poder leer de una forma más fácil los bancos actuales y poder leer nuevos próximamente.

El proceso metodológico empleado fue el siguiente:

1. Aprendizaje autónomo de *JavaScript y Electron*
2. Migración del código original en *Python* a *JavaScript*

3. Ejecución de pruebas funcionales con el nuevo lenguaje
4. Reestructuración y optimización del código para leer múltiples formatos de archivos bancarios
5. Implementación de pruebas adicionales para validar el desempeño de la aplicación
6. Validación de resultados con la empresa

2.2. Planeación y seguimiento del proyecto

- Descripción del proyecto

El proyecto consistió en transformar un código preexistente desarrollado en *Python* y *Streamlit*, en una aplicación local y visualmente amigable. Esta nueva versión permite realizar el rebalanceo de portafolios de inversión a partir de los archivos proporcionados de bancos y de los perfiles de riesgo que se manejan en la empresa.

Cada archivo bancario cuenta con una estructura distinta, por lo que fue necesario diseñar un sistema capaz de automatizar el análisis y procesamiento de los formatos que se manejan actualmente y de aquellos que pudieran incorporarse en el futuro. Asimismo, se previó que las categorías de inversión puedan ser modificadas —ya sea integradas o eliminadas— en los archivos actuales, sin generar conflictos en su lectura ni en el rebalanceo para no comprometer la funcionalidad del *Dashboard.*

Para afrontar este desafío, se implementó una nueva arquitectura, en la que se analizaron en detalle los archivos de cada banco, identificando similitudes entre las columnas relevantes y definiendo un mecanismo de "parseo" flexible que garantizara la correcta lectura de los datos.

El *Dashboard,* desarrollado a través de *JavaScript*, permite al usuario cargar los archivos del mes y de los bancos que desea, asignar su perfil de riesgo, y visualizar de manera clara y ordenada los movimientos de su portafolio. También, incluye recomendaciones para el rebalanceo (compra/venta de activos), advertencias sobre activos no reconocidos, y gráficas que reflejan la situación actual del portafolio.

El hecho de que la aplicación opere de forma local garantiza seguridad y confidencialidad de los datos financieros que se están tratando.

- Plan de trabajo

  Las actividades se organizaron de la siguiente manera:

  - Diagnóstico inicial:
    - *Actividades técnicas*: Aprendizaje autónomo de *JavaScript* y *Electron*.
    - *Actividades profesionales*: Reunión con los profesores para definir la estrategia de migración del código.
  - Desarrollo:
    - *Actividades técnicas*: Migración del código *Python* a *JavaScript*.
    - Actividades profesionales: Coordinación y trabajo colaborativo mediante *GitHub.*
  - Validación y pruebas:
    - *Actividades técnicas*: Pruebas funcionales del código para asegurar su correcto funcionamiento.
    - *Actividades profesionales*: Asesorías con profesores para resolver dudas específicas.
  - Visualización
    - *Actividades técnicas*: Reestructuración del código para automatizar la lectura de archivos bancarios a través de una sola función genérica.
    - *Actividades profesionales*: Asesorías adicionales y reunión de validación con un representante de la empresa.
  - Ajustes finales:
    - *Actividades técnicas*: Corrección de errores, ajustes visuales y aplicación de retroalimentación recibida.

        - *Actividades profesionales*: Redacción del RPAP y preparación de la presentación de la entrega final.

- Recursos utilizados:
    - *Humanos*: Equipo consolidado por 5 estudiantes de Ingeniería Financiera y 2 profesores.
    - *Tecnológicos*: Computadoras, JavaScript, CSS, HTML, Electron, GitHub, VisualCode.
    - *Tiempo*: 1 mes y 1 semana.
    - Asesorías.
- Desarrollo de la propuesta de mejora:

A lo largo del desarrollo del proyecto se implementaron funcionalidades adicionales que representaron mejoras reales y significativas para la empresa, tales como:

- *Advertencias*: cada activo es comparado con un catálogo consolidado predefinido, si no se encuentra una coincidencia, se clasifica automáticamente como “NO RECONOCIDO” y se lanza una advertencia que aparece en la pantalla al usuario y le indica qué activo de qué no pudo ser clasificado.
- *Indicador de capital pendiente*: Si hay activos no reconocidos en la categorización, al mostrar el valor total de portafolio, no se está considerando el valor de los activos no considerados, pero, abajo aparece un capital pendiente a categorizar, para que el usuario sea consciente de la cantidad que falta en ese total.
- *Comparaciones*: Al ingresar, tiene la opción de cargar un archivo en un apartado llamado comparaciones, que después aparecerá la información resumida y limpia en un apartado llamado de la misma forma, lo cual le sirve para ver los cambios que tuvo de un mes a otro en un banco.
- *Advertencia de formato de archivo no reconocido*: Si se carga un nuevo archivo de banco el cual no pudo ser procesado, la aplicación le indica al usuario qué columna debe renombrarse (ya sea la del nombre del activo

o la del valor de mercado) para que el archivo pueda ser aceptado por el sistema tras una pequeña modificación manual.

Para ejemplificar la lógica y estructura del código, en el Anexo 2 se muestra la implementación de la clase *Banks,* que representa la estructura base para la lectura y gestión de archivos de entrada de los distintos bancos.

En el Anexo 3, se presenta el código del archivo *parser*, encargado de la limpieza y estandarización de nombres de las columnas, además de la eliminación de datos irrelevantes, garantizando uniformidad en las estructuras de datos.
Además, en el Anexo 4 se muestra el código *rebalancer,* donde se realiza la categorización automática. También, se realiza una comparativa entre las descripciones de los activos y un archivo auxiliar de categorías. Si no se logra encontrar una coincidencia, los activos son categorizados como "No reconocido".

Primeramente, al abrir la aplicación se muestra la página de inicio mostrando el logo de la empresa de Impulso Patrimonial, para poder acceder al *Dashboard* se solicita el correo y contraseña al usuario. (Figura 1)

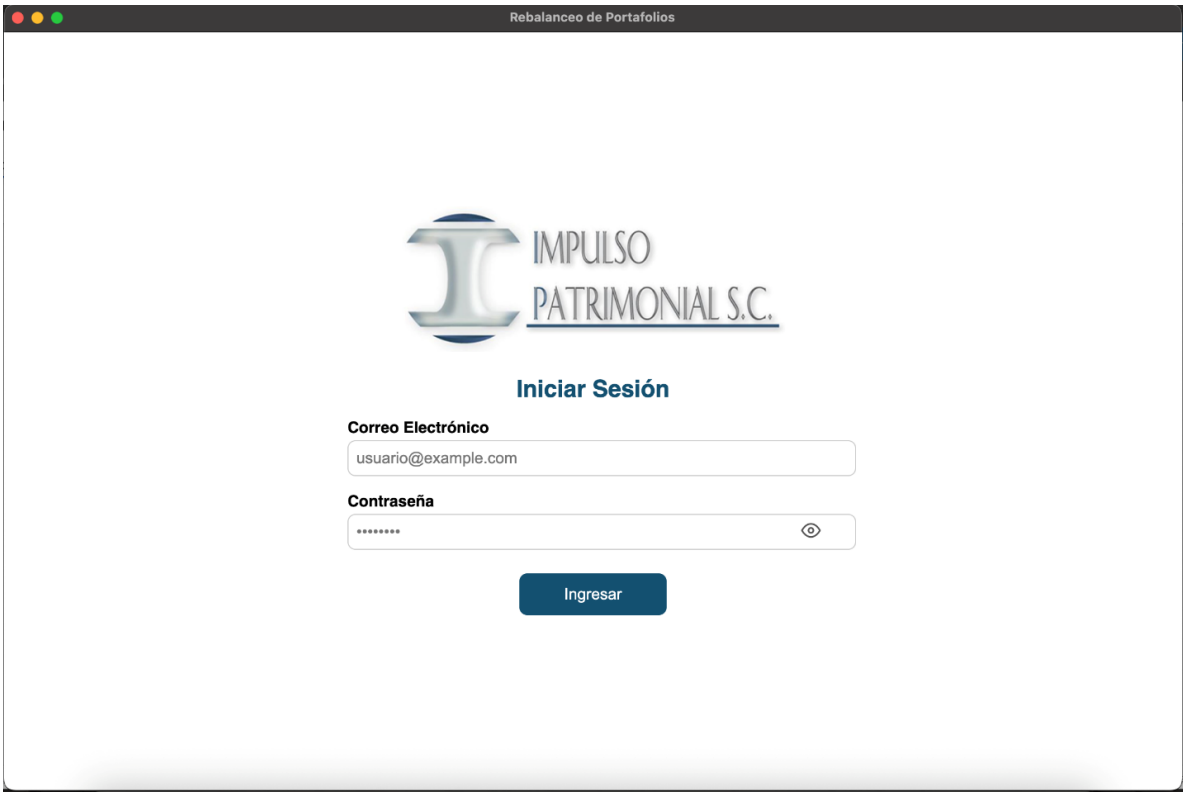


Figura 1. Página de inicio.

Posterior al inicio de sesión, se muestra la siguiente pantalla (Figura 2) en la cual, se puede visualizar el logo de la empresa nuevamente, seguido por un menú interactivo en el *sidebar,* dónde se cargan los archivos de bancos, el archivo de perfiles de riesgo, archivo de categorías y los archivos para generar un comparativo (este último rubro siendo opcional), finalmente despliega las opciones de perfil de riesgo.

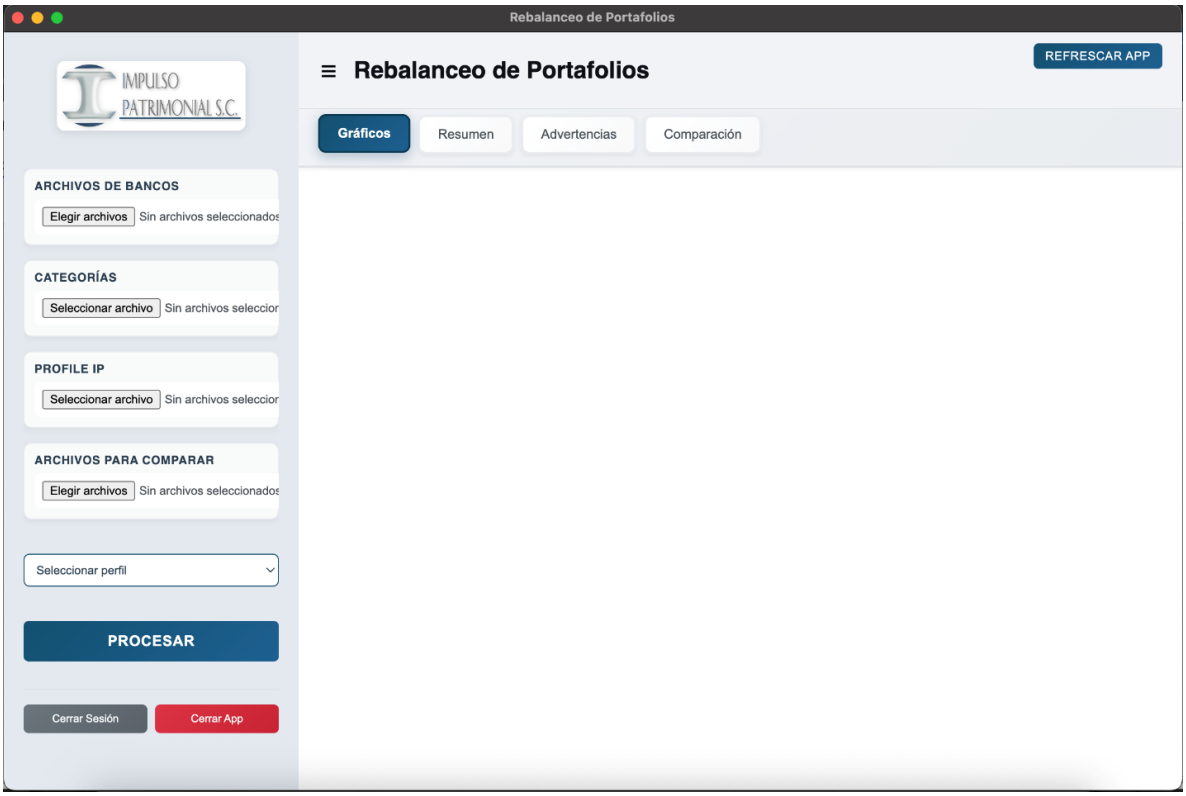


Figura 2. Página principal del Dashboard

Una vez iniciada la sesión, se realiza la carga de archivos y se da clic en el botón 'Procesar' para activar el proceso de información y generar las distintas visualizaciones de la data que genera el *Dashboard.* La página cuenta con cuatro secciones: Gráficos, Resumen, Advertencias y Comparación. Primeramente, la sección de *Gráficos* genera 3 tipos de gráficos:

1. **Gráfico de barras**: Compara los valores de mercado actual que se tienen y los valores objetivo según el perfil del usuario por categoría. Este gráfico permite identificar en qué categorías hay subinversión o sobreinversión (Figura 3).
2. **Gráfico de barras divergentes:** Muestra las operaciones por categoría e indica los montos a comprar (verde) y a vender (rojo) por categoría después de realizar el rebalanceo. Este formato divergente permite visualizar intuitivamente qué categorías tienen mayor desbalance y la acción a tomar (Figura 4).

3. **Gráficos circulares:** Se muestran dos gráficos circulares (*pie chart*), del lado izquierdo se muestra la distribución real actual del portafolio y del lado derecho, los porcentajes recomendados a invertir por categoría de instrumentos según el perfil de riesgo (Figura 5).

   Asimismo, el *Dashboard* permite visualizar los gráficos mencionados agrupados por todas las categorías o bien por las subcategorías de renta variable o renta fija. La visualización se puede seleccionar utilizando la lista que se desliza en la parte superior del gráfico de Comparación de Valores de Mercado.

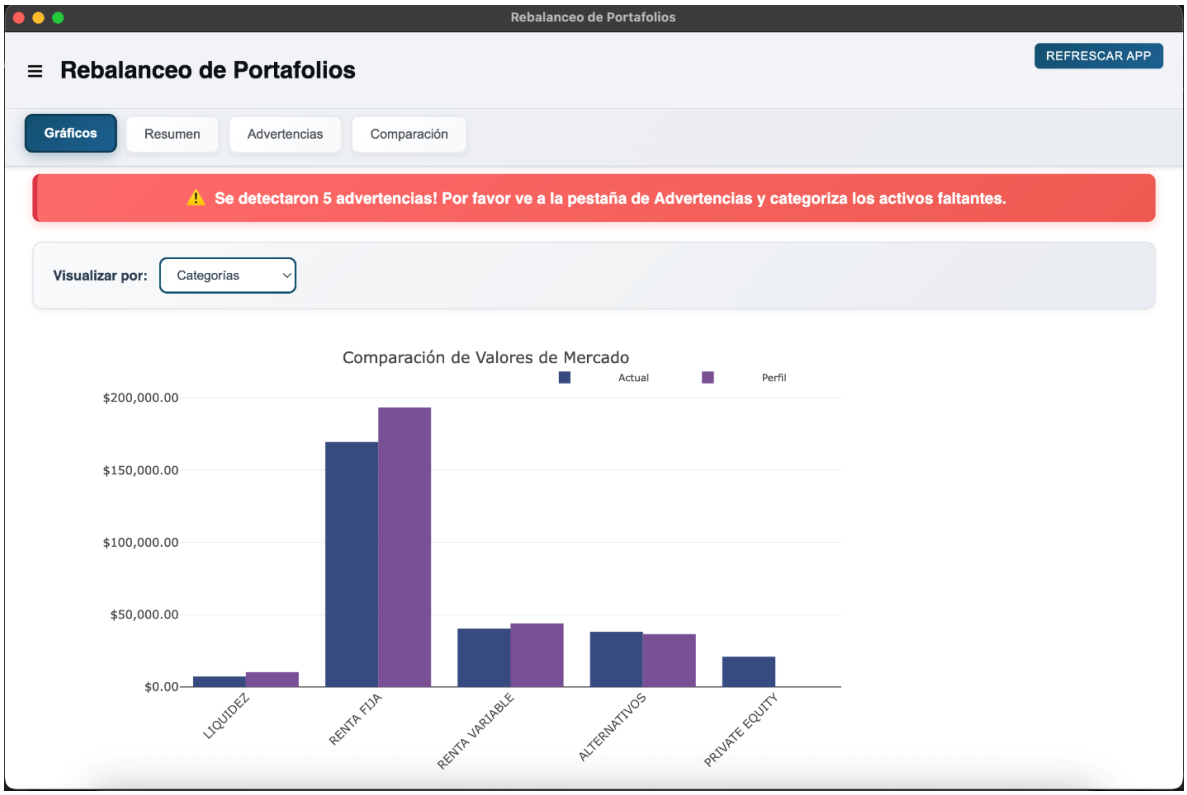


Figura 3. Gráfico de barras verticales.[1]

[1] **Nota:** Los datos mostrados son falsos, por confidencialidad no presentan la información de ninguna cartera de inversión.

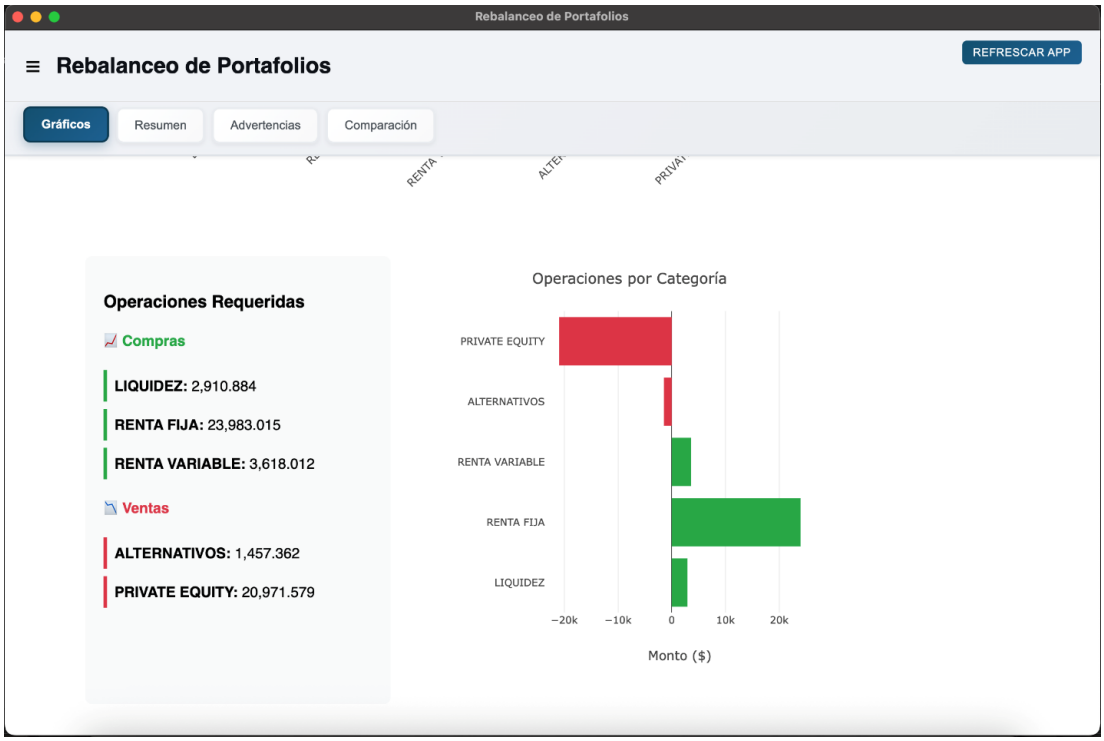


Figura 4. Gráfico de barras divergentes[2].

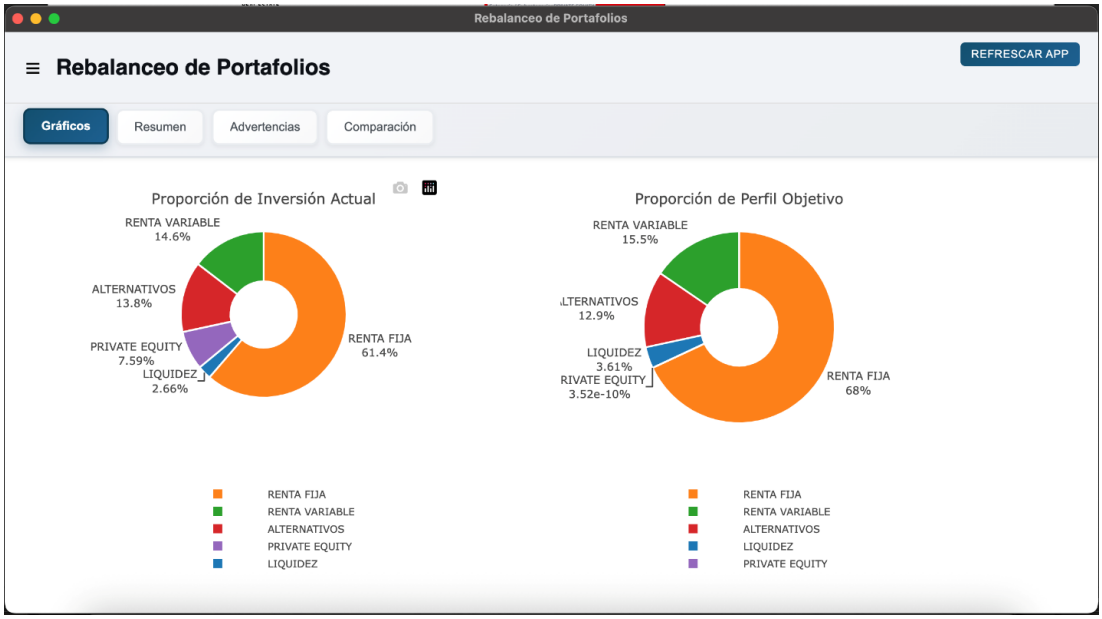
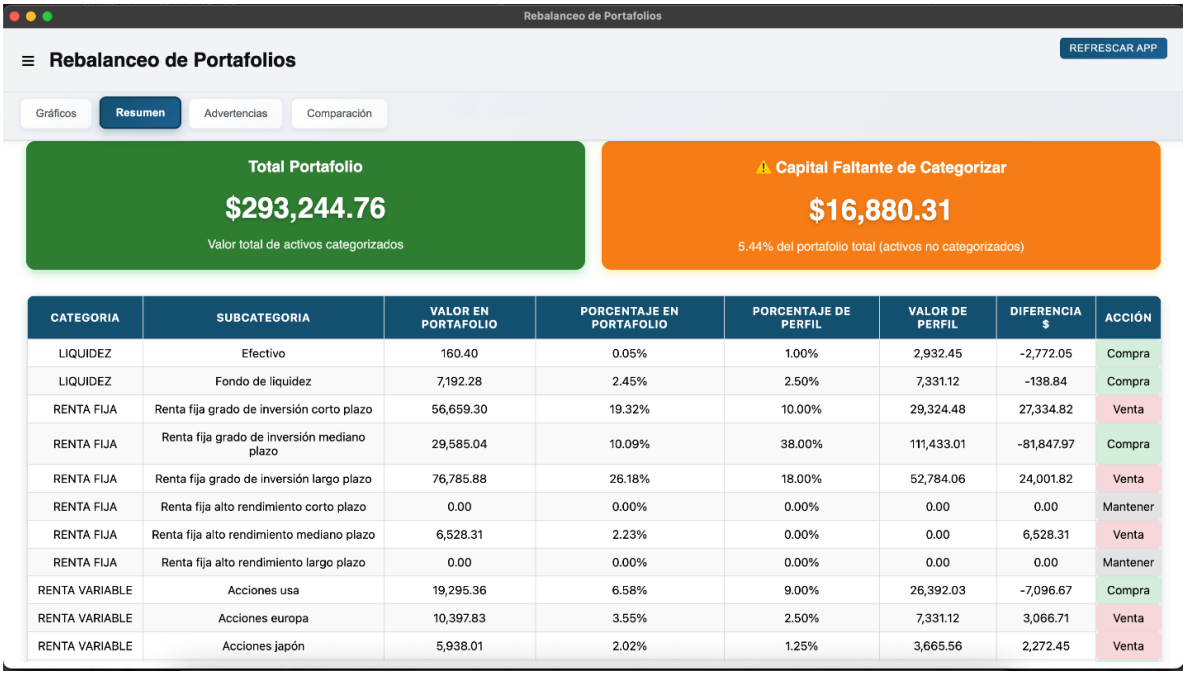


Figura 5. Comparativo de gráficos de pastel.[3]

[2] **Nota:** Los datos mostrados son falsos, por confidencialidad no presentan la información de ninguna cartera de inversión.

[3] **Nota:** Los datos mostrados son falsos, por confidencialidad no presentan la información de ninguna cartera de inversión.

En la sección de *Resumen*, se muestra un reporte consolidado de toda la información procesada de los archivos cargados en la sección de Archivos de Bancos (Figura 6). La tabla resultante muestra por categoría y subcategoría los valores actuales de mercado de cada activo del portafolio y su variación con el porcentaje determinado para el perfil de riesgo del cliente, así como la acción a tomar para ajustar el portafolio a su composición óptima.

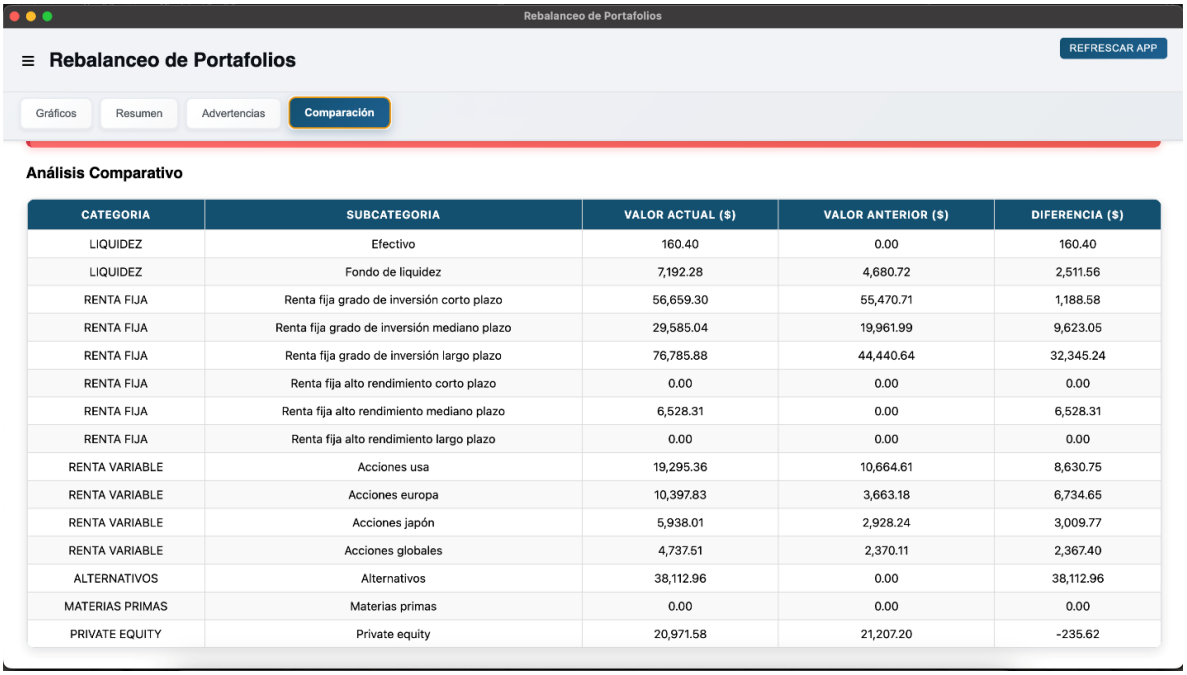


CATEGORIA	SUBCATEGORIA	VALOR EN PORTAFOLIO	PORCENTAJE EN PORTAFOLIO	PORCENTAJE DE PERFIL	VALOR DE PERFIL	DIFERENCIA $	ACCIÓN
LIQUIDEZ	Efectivo	160.40	0.05%	1.00%	2,932.45	-2,772.05	Compra
LIQUIDEZ	Fondo de liquidez	7,192.28	2.45%	2.50%	7,331.12	-138.84	Compra
RENTA FIJA	Renta fija grado de inversión corto plazo	56,659.30	19.32%	10.00%	29,324.48	27,334.82	Venta
RENTA FIJA	Renta fija grado de inversión mediano plazo	29,585.04	10.09%	38.00%	111,433.01	-81,847.97	Compra
RENTA FIJA	Renta fija grado de inversión largo plazo	76,785.88	26.18%	18.00%	52,784.06	24,001.82	Venta
RENTA FIJA	Renta fija alto rendimiento corto plazo	0.00	0.00%	0.00%	0.00	0.00	Mantener
RENTA FIJA	Renta fija alto rendimiento mediano plazo	6,528.31	2.23%	0.00%	0.00	6,528.31	Venta
RENTA FIJA	Renta fija alto rendimiento largo plazo	0.00	0.00%	0.00%	0.00	0.00	Mantener
RENTA VARIABLE	Acciones usa	19,295.36	6.58%	9.00%	26,392.03	-7,096.67	Compra
RENTA VARIABLE	Acciones europa	10,397.83	3.55%	2.50%	7,331.12	3,066.71	Venta
RENTA VARIABLE	Acciones japón	5,938.01	2.02%	1.25%	3,665.56	2,272.45	Venta

Figura 6. Dataframe de Reporte consolidado.[4]

En seguida, la sección de *Comparación* muestra contenido únicamente en caso de haber cargado archivos en la parte correspondiente a Archivos para Comparar. Esta sección genera un resumen del valor de mercado de los archivos cargados en la sección de Archivos de Banco y otro resumen de los archivos a comparar, ambos agrupados por categoría y subcategoría, mostrando la diferencia en dinero. (Figura 7).

[4] **Nota:** Los datos mostrados son falsos, por confidencialidad no presentan la información de ninguna cartera de inversión.



CATEGORIA	SUBCATEGORIA	VALOR ACTUAL ($)	VALOR ANTERIOR ($)	DIFERENCIA ($)
LIQUIDEZ	Efectivo	160.40	0.00	160.40
LIQUIDEZ	Fondo de liquidez	7,192.28	4,680.72	2,511.56
RENTA FIJA	Renta fija grado de inversión corto plazo	56,659.30	55,470.71	1,188.58
RENTA FIJA	Renta fija grado de inversión mediano plazo	29,585.04	19,961.99	9,623.05
RENTA FIJA	Renta fija grado de inversión largo plazo	76,785.88	44,440.64	32,345.24
RENTA FIJA	Renta fija alto rendimiento corto plazo	0.00	0.00	0.00
RENTA FIJA	Renta fija alto rendimiento mediano plazo	6,528.31	0.00	6,528.31
RENTA FIJA	Renta fija alto rendimiento largo plazo	0.00	0.00	0.00
RENTA VARIABLE	Acciones usa	19,295.36	10,664.61	8,630.75
RENTA VARIABLE	Acciones europa	10,397.83	3,663.18	6,734.65
RENTA VARIABLE	Acciones japón	5,938.01	2,928.24	3,009.77
RENTA VARIABLE	Acciones globales	4,737.51	2,370.11	2,367.40
ALTERNATIVOS	Alternativos	38,112.96	0.00	38,112.96
MATERIAS PRIMAS	Materias primas	0.00	0.00	0.00
PRIVATE EQUITY	Private equity	20,971.58	21,207.20	-235.62

Figura 7. Sección de Archivos Comparativos.[5]

Finalmente, el Dashboard cuenta con una sección de Advertencias en la cual aparece el desglose de activos presentes en los archivos de bancos que no pudieron categorizarse (Figura 8). En el desglose aparece el nombre del activo, del archivo y el mes correspondiente, de modo que sea fácil identificarlos y clasificarlos manualmente actualizando el archivo de categorías '*categories.csv*'. El resto de las secciones muestra una alerta de advertencias encontradas con la finalidad que el usuario revise esta sección con el listado completo de advertencias. En caso de no haber advertencias las secciones restantes no mostrarán la alerta y esta sección tendrá un texto especificando que no se encontraron advertencias.

[5] **Nota:** Los datos mostrados son falsos, por confidencialidad no presentan la información de ninguna cartera de inversión.

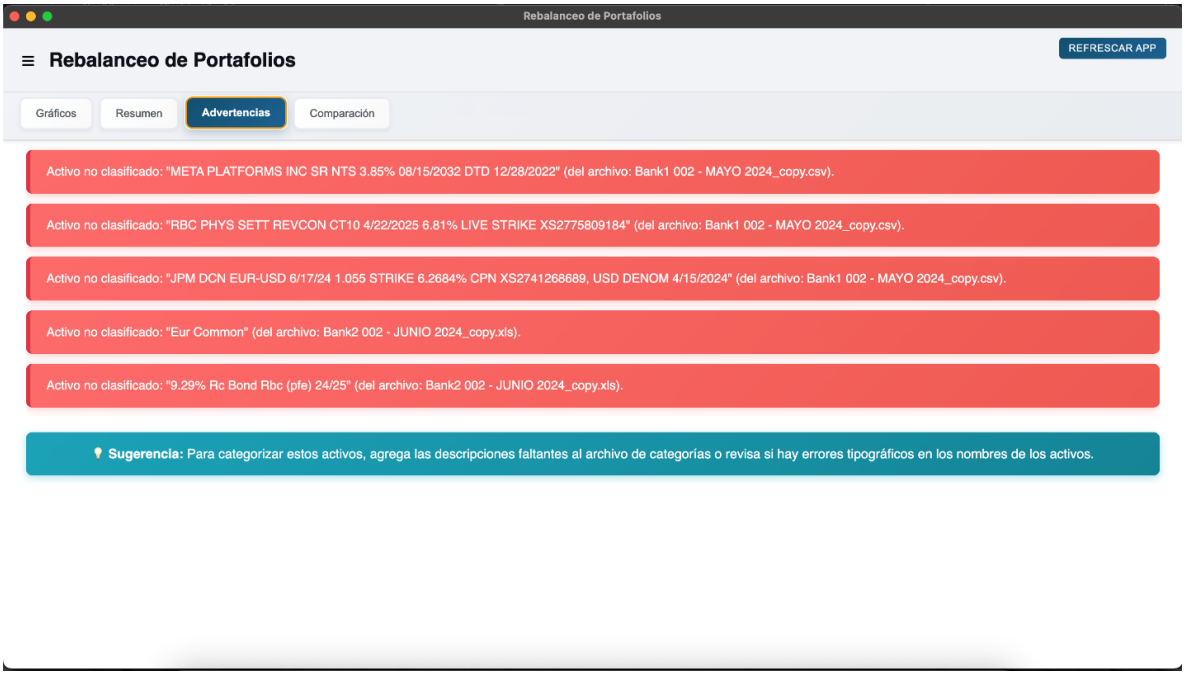


Figura 8. Advertencias.

3. Resultados del trabajo profesional

El principal resultado del proyecto fue el desarrollo de una aplicación local completamente funcional, adaptada a las necesidades reales que tiene Impulso Patrimonial. Esta herramienta permite subir archivos de los cinco bancos con los que actualmente trabajan, e incluso nuevos, interpretar correctamente su contenido, consolidar la información relevante y generar un *Dashboard* interactivo con visualizaciones claras, así como recomendaciones de compra o venta personalizadas para su rebalanceo.

La aplicación funciona de forma local, sin requerir conexión a internet ni servidores externos. Cuenta con una pantalla de inicio que solicita correo y contraseña para restringir el acceso sólo las personas autorizadas; la interfaz es intuitiva y fácil de navegar, incluye advertencias visibles que guían al usuario en caso de inconsistencias, tablas organizadas, gráficas interactivas, indicadores del valor total del portafolio y del capital pendiente por categorizar y comparaciones con un portafolio ideal según el perfil de riesgo asignado para el cliente.

Uno de los logros más significativos fue la automatización exitosa de un proceso que anteriormente se realizaba de forma manual dentro de la empresa, con alto riesgo de errores y una inversión considerable de tiempo. La categorización automática de activos, junto con la detección de los no reconocidos, permite una mejora continua del sistema y una mayor precisión en la toma de decisiones.

El *Dashboard* desarrollado, facilita la comprensión del estado actual del portafolio en relación con el perfil de riesgo del inversionista, generando recomendaciones accionables que pueden ser aplicadas directamente por la empresa.

Además de los resultados técnicos, se fortalecieron habilidades profesionales indispensables como el trabajo colaborativo en *GitHub*, el aprendizaje autónomo de un nuevo lenguaje de programación, realización de pruebas con archivos reales y la experiencia directa con un usuario real.

Se entregó una solución personalizada que responde a una necesidad concreta de una empresa, con un impacto directo en su flujo operativo mensual de trabajo; además, de contribuir indirectamente a mejorar la salud financiera de los clientes de Impulso Patrimonial facilitando la toma de decisiones más informadas sobre sus inversiones.

Finalmente, el proyecto representó una oportunidad para aplicar de manera integral los conocimientos adquiridos durante la carrera en un proyecto real, con desafíos reales; además de que se aprendieron cosas extras a las vistas en la carrera y se pudieron presentar los resultados a una empresa real. El resultado fue bien recibido por Impulso Patrimonial, lo que demuestra la capacidad del equipo para abordar problemáticas de la vida real en el sector financiero.

4. Reflexiones del alumno o alumnos sobre sus aprendizajes, las implicaciones éticas y los aportes sociales del proyecto

Reflexión Karimme Anahi Tejeda Olmos

***Aprendizajes* Profesionales**

Durante este proyecto logré desarrollar varios aprendizajes del ámbito profesional, iniciando por la implementación de un proyecto real que atendía necesidades reales dentro del sistema financiero, por lo que mi capacidad del planteamiento para la solución del problema debía de ser realizada con total seriedad y profesionalismo para poder entregar un buen proyecto que resolviera lo que se pedía.

Otro aprendizaje, fue mejorar el trabajo en equipo, lo cual siempre ha sido una debilidad en mí, porque no sé delegar y siempre quiero hacer todo yo, además, de que no reconozco cuando requiero de ayuda; pero durante el PAP, mejoré en la repartición de actividades y pude pedir apoyo a mis compañeros o profesores.

Mejoré el uso de *GitHub*, ya que es una herramienta importante en el ámbito profesional y se volvió indispensable a lo largo del semestre. Aprendí bastante el lenguaje de *JavaScript*, *CSS* y *HTML* y la creación de aplicaciones por medio de *Electron*.

En cuanto a los aprendizajes obtenidos para mi vida profesional, puedo destacar la experiencia obtenida al ser mi primer acercamiento a una empresa financiera; también logré ampliar y explorar el área de la programación aplicada en el sector financiero y bursátil, lo cual me gustó demasiado y me abrió la visión para mis opciones al elegir mi maestría y trabajo.

Aprendizajes Sociales

Una de nuestras principales ideas de transformación fue que la aplicación que creamos pudiera ser de fácil acceso y entendible para cualquiera persona, por lo que quisimos agregar gráficas, colores, un menú que te pudiera ir direccionando a dónde dirigirte y te

pudiera hacer más fácil el proceso de rebalanceo del portafolio, lo cual va a generar un impacto en el ahorro de tiempo de la empresa y en el poder evitar los errores humanos, conduciendo a que se tomen mejores decisiones financieras para las personas que están invirtiendo, lo cual les ayudará a cuidar sus ahorros y a hacerlos crecer.

Esto lo considero muy importante porque se debe de mantener la confianza en el pequeño grupo de inversionistas mexicanos actuales que ya dieron el primer paso de comenzar a invertir, para poder lograr que este grupo vaya creciendo con confianza y con conocimientos financieros.

Aprendizajes Éticos

Después de esta experiencia PAP quiero continuar apostando por empresas éticas mexicanas y por los mexicanos, no necesito irme al extranjero para aplicar mi carrera como Ingeniera Financiera, sé que hay muchas cosas por hacer en México.

El acceso a la educación es un privilegio al cual muy pocos tenemos acceso y debemos de compartir nuestros conocimientos con quiénes más podamos para ayudarnos a crecer todos juntos, cada persona podemos aprenderle mucho a otra persona y enriquecernos de sus conocimientos.

Aprendizajes en lo personal

Este PAP fue un reto para mí en todos los sentidos, pero hoy puedo reconocer que logré cumplir con el objetivo esperado satisfactoriamente, aprendí demasiado, me enriquecí mucho de los conocimientos de mis profesores, pero, especialmente de mis compañeros de equipo con los que estuve conviviendo todo el semestre. Son unas personas muy inteligentes, estudiosas y capaces, a las cuales les admiro por sus habilidades de resolución, por sus conocimientos y por su manejo del tiempo.

Reflexión Íñigo de Obeso Landeros

Aprendizajes profesionales

Durante este PAP consolidé competencias técnicas y metodológicas clave para el desarrollo de soluciones *front-end* en un entorno real. Aprendí a combinar *JavaScript, HTML y CSS* para crear un *dashboard* de rebalanceo de portafolios que responde a las necesidades del *stakeholder.* La migración de la lógica de *Python/Streamlit* a *Electron* me obligó a profundizar en arquitectura de aplicaciones de escritorio y empaquetado de recursos. Adicionalmente, mejoré mis buenas prácticas de programación orientada a objetos, control de versiones en *GitHub* y despliegue de *builds* para que cualquier usuario pueda instalar sin usar la terminal. Esta experiencia profesional elevó mi capacidad para entregar software robusto, modular y de fácil mantenimiento.

Aprendizajes Sociales:

Al diseñar la interfaz y la experiencia de usuario comprendí la importancia de hacer accesible y comprensible la información financiera a distintos perfiles. Trabajar en equipo me enseñó cómo la empatía y la escucha activa mejoran la usabilidad de la aplicación, pues cada sugerencia de diseño o *feedback* de mis compañeros aportó a una solución más inclusiva y colaborativa.

Aprendizajes Éticos:

Este proyecto me hizo reflexionar sobre el compromiso que implica crear software financiero: cada línea de código que procesa datos de inversión conlleva una responsabilidad ética. Asimismo, las prácticas de documentación claras permiten que otros estudiantes y profesionales en México puedan replicar y mejorar la aplicación sin barreras de acceso. Con esto, reafirmé mi convicción de impulsar a entidades financieras a optimizar sus procesos, así, haciendo más escalables sus operaciones y brindando mayores oportunidades de inversión a la población mexicana.

Aprendizajes Personales:

En lo personal, enfrentar el reto de aprender *Electron* y todo el *stack front-end* en un tiempo ajustado puso a prueba mi adaptación y mi capacidad de autoaprendizaje. Descubrí que soy capaz de asimilar nuevos *frameworks* y librerías bajo presión, y que delegar tareas según fortalezas del equipo optimiza los plazos sin sacrificar la calidad. Además, reforcé mi comunicación asertiva y comprendí el valor de equilibrar autonomía con colaboración, permitiéndome crecer tanto en confianza como en resiliencia frente al resolver problemas.

Reflexión Daniel Sánchez López

***Aprendizajes* Profesionales**

Este proyecto representó un desafió significativo porque tuve que adaptarme de forma rápida para aprender a usar herramientas y lenguajes de programación que nunca había requerido por el contexto de la carrera de ingeniería financiera, sin embargo. Gracias a este proyecto me di cuenta de la capacidad que tengo para adaptarme a estos requerimientos, a trabajar eficientemente en equipo y cumplir satisfactoriamente con el desarrollo de la aplicación.

Aprendí la importancia de la versatilidad técnica, para aportar valor a un proyecto en el cuál cambian los requerimientos y necesidades del cliente. Además, otro aprendizaje que me gustaría resaltar fue el de Project Management, gracias a este PAP pude aprender métodos y técnicas de gestionar proyectos a nivel empresarial, que fueron claves para dividir el proyecto en tareas granulares, asignar prioridades y estimar tiempos, lo que facilitó el control del proyecto y nos facilitó la entrega del primer *release* de la aplicación. Este proyecto refuerza a necesidad de una planificación estructurada y la importancia de la integración de la tecnología en roles de finanzas.

Aprendizajes Sociales

El proyecto resaltó la dinámica colaborativa en el equipo, donde la comunicación y apoyo mutuo durante todo el proyecto impulsó el desarrollo de la aplicación, aun cuando la mayoría del equipo, que estuvo conformado únicamente de ingenieros financieros, no tenía

un *background* técnico o de desarrollo de aplicaciones. La presión del tiempo limitado nos enseñó a dividir la carga de trabajo y también a pedir ayuda y a desarrollar esa colaboración y apoyo mutuo dentro del equipo. Asimismo, las interacciones con el cliente nos mostraron la importancia de alinear expectativas y retroalimentación con nuestro trabajo, que me poco a poco me ayuda a hacer *networking* y me reafirma la importancia de la colaboración en entornos profesionales.

Aprendizajes Éticos

Éticamente, el desarrollo de esta aplicación me dio la responsabilidad de manejar correctamente datos sensibles de clientes, además de que me impulsó a asegurar que el funcionamiento de la aplicación fuera correcto y los cálculos y operaciones que realiza sean precisos sin mostrar errores. El manejo del patrimonio de las personas es un punto crítico y sensible al desarrollar una herramienta que agilice el proceso de la gestión de ese patrimonio. Este proyecto reforzó el compromiso ético con la excelencia y la responsabilidad en el desarrollo de herramientas que afectan el patrimonio de terceros.

Aprendizajes en lo personal

Yo aprendí bastante en este periodo de verano durante el trabajo del PAP, reafirmé mi capacidad de adaptación y de resiliencia bajo presión, y mi capacidad de trabajar en equipo y de relacionarme y aprender de cada uno de mis compañeros. Me doy cuenta de las herramientas y aprendizajes que adquirí gracias a este proyecto y que puedo llevar a otros entornos o ambientes laborales, el valor que puedo aportar en cualquier proyecto u organización.

También me motivó a ser proactivo y a estar en constante comunicación con mi equipo para dar seguimiento a las actividades y tareas y asegurar de que la aplicación funcionara y no tuviera errores. Este proyecto me permitió colaborar y aprender muchos temas que me ayudan personalmente en mi crecimiento y desarrollo financiero, así como habilidades que puedo ponerlas en práctica en asuntos y situaciones tanto personales como profesionales. Por último, agradezco mucho la oportunidad de trabajar con mi equipo, demostramos

nuevamente nuestra capacidad y nuestro compromiso de trabajo, fue una grata experiencia y es una gran sorpresa que me llevo, que gracias a mis compañeros la experiencia de mis dos PAP fue mucho mejor de lo que esperaba e hicieron un gran momento de mi carrera en ITESO.

Reflexión Alberto Marín Gómez

***Aprendizajes* Profesionales**

Este proyecto me hizo aprender mucho sobre como trabajar en equipo sobre un mismo código de manera más profesional y ordenada, mis compañeros me aportaron mucho para poder aprender a hacer un *git, pull y commit.* También sobre cómo debemos estructurar un proyecto a largo plazo y como estructurar en un diagrama las diferentes tareas que debemos cumplir para llegar a los objetivos establecidos.

Aprendizajes Sociales

Este *proyecto* demuestra la importancia de un ingeniero financiero en el desarrollo de programas que puedan automatizar estos procesos para bajar los costos de rebalanceo de portafolio y que los productos financieros se puedan volver más accesibles para todo público.

Aprendizajes Éticos

Considero que para este proyecto es muy importante tener la parte ética sobre la responsabilidad con los datos que manejamos ya que son de una empresa que nos la están proporcionando y la divulgación de los datos podría afectar gravemente a las personas ya sean clientes o trabajadores. También la importancia de siempre estar para el equipo y nunca dejar de ver en que aportar. Y cumplir con las tareas establecidas en tiempo y forma para no perjudicar al compañero que tiene tareas que dependen de las tareas que tú tienes.

Aprendizajes en lo personal

Este proyecto me hizo reflexionar sobre las herramientas que tengo para este proyecto y me demostró que aún me falta obtener nuevos conocimientos para poder desarrollar una aplicación. Considero que es algo bueno el descubrir que aún tenemos que aprender nuevas habilidades y que creo que cuando las obtenga podré construir otro tipo de herramientas financieras.

Reflexión Marianne Trujillo Altamirano

***Aprendizajes* Profesionales**

Integrarme a este proyecto representó un reto significativo desde el inicio, principalmente porque se trataba de una continuación de un trabajo que ya se venía desarrollando desde el semestre anterior. Esto me exigió aprender rápidamente sobre herramientas y lenguajes que no dominaba, como *Streamlit* y *JavaScript*, además de comprender a profundidad un repositorio ya establecido.

Por otro lado, esta experiencia fue valiosa porque anteriormente había trabajado con *PyCharm* y *Git Bash,* pero *VS Code* me pareció más práctico e intuitivo, específicamente cuando se trabaja en equipo y con control de versiones. Además, aprendí a gestionar mejor mis ramas y *commits* en *GitHub*, lo cual reforzó mis habilidades en trabajo colaborativo en entornos de desarrollo reales.

Una de las cosas que más valoro es que aprendí a leer y modificar código que no escribí yo. Comprender cómo estaba estructurado el repositorio y adaptarme a los estilos de programación previos me ayudó a desarrollar una mayor flexibilidad técnica y profesional.

Aprendizajes Sociales

Este proyecto me permitió integrarme a un equipo que ya tenía una dinámica establecida, lo cual representó un desafío importante. Desde el primer momento, mis compañeros que habían trabajado previamente en el proyecto mostraron una gran disposición para explicarme conceptos, compartir documentación y facilitar mi incorporación.

Por mi parte, me comprometí a colaborar activamente, a escuchar activamente y a buscar la manera de sumar desde mis propias fortalezas. Esta experiencia me enseñó la importancia de una comunicación clara, constante y empática dentro de un equipo de desarrollo. Aprendí también que saber pedir ayuda no solo es válido, sino necesario para avanzar con mayor eficiencia y asegurar la calidad del trabajo.

La buena convivencia y la proactividad nos permitieron repartir responsabilidades estratégicamente, aprovechando lo mejor de cada persona. Esto también fortaleció el sentido de compromiso colectivo, ya que sabíamos que los avances de uno impactaban directamente en los logros del grupo. Fue muy motivador ver cómo, con un objetivo claro y una estructura de trabajo colaborativa, es posible lograr grandes avances en poco tiempo.

Aprendizajes Éticos

Durante el desarrollo del proyecto, uno de los temas más importantes que aprendí a considerar fue la responsabilidad en el manejo de datos sensibles, ya que trabajamos con información confidencial. Esto me hizo más consciente de la necesidad de proteger la privacidad de la información, tanto en la lógica del código como en las presentaciones o el uso de herramientas externas.

Tuve que tomar decisiones éticas sobre qué información se podía compartir al buscar soluciones como el utilizar inteligencia artificial, o incluso al generar ejemplos para las interfaces del *dashboard.* Este ejercicio de conciencia ética me permitió reforzar la importancia de desarrollar software con enfoque responsable y profesional.

Además, procuré mantener la transparencia en todo momento, asegurándome de que cada avance técnico cumpliera con el objetivo final del proyecto: entregar una solución funcional y accesible para el *stakeholder,* que pudiera ser ejecutada sin complicaciones, a diferencia de la versión anterior que requería múltiples instalaciones y configuraciones en terminal.

Aprendizajes en lo personal

Esta experiencia me ayudó a confirmar que tengo la capacidad de adaptarme a entornos complejos y a proyectos en marcha. Fue muy significativo darme cuenta de que, aunque al

principio no dominara todas las herramientas necesarias, con dedicación, apoyo de mi equipo y un buen plan de estudio, fui capaz de integrarme y contribuir activamente al desarrollo de la aplicación.

Aprendí también a gestionar mi tiempo y mi energía, ya que tuve que balancear el aprendizaje autónomo con las responsabilidades asignadas. Esta experiencia me dejó claro que el trabajo colaborativo bien estructurado puede potenciar las capacidades individuales, y que estar rodeada de un equipo comprometido puede hacer toda la diferencia. También descubrí que disfruto mucho trabajar en la creación de soluciones tecnológicas, especialmente cuando tienen un impacto directo en facilitar procesos reales para otras personas o empresas.

5. Conclusiones

Al concluir el PAP y poder visualizar los objetivos planteados y esperados al inicio del semestre con los resultados que obtuvimos, se lograron resultados bastante satisfactorios que superaron lo esperado tanto por los profesores, por la empresa y por los integrantes del equipo.

Desde el comienzo del verano, se tuvo una organización excelente en la que la repartición de actividades permitió a todos los integrantes trabajar y cumplir en tiempo con lo que le correspondía, se realizó cada entrega parcial en tiempo y forma, logrando tener una calificación de 10 sobre 10 en cada una de las entregas parciales.

Durante el verano, se tuvo una reunión con un representante de la empresa, en la cual se le presentó la aplicación creada y quedó totalmente satisfecho con los resultados.

Se logró crear una aplicación local en *Electron* y entregar los ejecutables en formatos ***.exe*** y ***.dmg*** para los diferentes sistemas operativos (*Windos y McOs* respectivamente), los cuales son bastante amigables con el usuario, al igual que el *dashboard,* que contiene colores, gráficas, tablas, de los 5 bancos actuales y puede recibir nuevos bancos; además de que se tiene un control al ingresar a la página que requiere de usuario y contraseña.

La aplicación puede tener mejoras constantes para seguir beneficiando a Impulso Patrimonial, el código de la aplicación cuenta con una buena organización y estructura que permite ser ampliado y modificado en caso de que se requieran mejoras para esta aplicación. Una de las ideas de mejoras que tenemos, es conectarlo por medio de una API para que pueda estar recibiendo las últimas noticias sobre los activos financieros en su pantalla principal y poder tomar decisiones en sus portafolios con base a estas.

El equipo está sumamente satisfecho con su trabajo, la participación del equipo fue equitativa, el ambiente fue excelente y se logró trabajar sin inconvenientes durante todo el semestre, lo cual se refleja en el proyecto entregado.

6. Bibliografía

Zhang, Y. Z., Ahluwalia, H. A., Ying, A. Y., Rabinovich, M. R., & Geysen, A. G. (2022). Rational Rebalancing: an analytical approach to multiasset portfolio rebalancing decisions and insights. *Vanguard*.

Garay, F. & Escuela Directa. (s. f.). *JavaScript TOTAL - de cero a programador web en 18 días*. UDEMY. https://www.udemy.com/course/javascript-total/?couponCode=MT150725B

Build cross-platform desktop apps with JavaScript, HTML, and CSS | Electron. (s. f.). https://www.electronjs.org/

Node.js — Run JavaScript everywhere. (s. f.). https://nodejs.org/es

Building your First App | Electron. (s. f.). https://www.electronjs.org/docs/latest/tutorial/tutorial-first-app

¿Por qué Electron? | Electron. (s. f.). https://www.electronjs.org/es/docs/latest/por-qu%C3%A9-electron

Anexos (en caso de ser necesarios)

Anexo 1. Main.js

```
// src/main/main.js
console.log("--- CARGANDO: main.js ---");
const { app, BrowserWindow, ipcMain } = require('electron');
const path = require('path');
const {runFullAnalysis } = require('./processor.js');
const { readCsv, readProfileData } = require('./fileUtils.js');

function createWindow() {
  const mainWindow = new BrowserWindow({
    width: 1200,
    height: 800,
    webPreferences: {
      preload: path.join(__dirname, './preload.js'),
      contextIsolation: true,
      nodeIntegration: false,
    }
  });

  mainWindow.loadFile(path.join(__dirname, '../renderer/index.html'));
```

```
}

app.whenReady().then(createWindow);
app.on('window-all-closed', () => { if (process.platform !== 'darwin') app.quit() });

ipcMain.handle('app:run-analysis', async(event, payload) => {
  console.log("[Main] Solicitud 'app:run-analysis' recibida. Delegado a processor.js...");
  return await runFullAnalysis(payload);
});

ipcMain.handle('files:readCsv', (e, filePath) => readCsv(filePath));

ipcMain.handle('files:readProfile', (e, filePath) => readProfileData(filePath));

ipcMain.handle('portfolio:process', async (e, data) => {
  console.log("[Main] Solicitud 'portfolio:process' recibida. Delegando a processor.js...");
  return await processAndRebalance(data);
});

ipcMain.on('close-app', () => {
  app.quit();
});

ipcMain.on('logout', () => {
  mainWindow.webContents.send('force-logout');
});
```

Anexo 2. Banks.js

```
// src/main/banks/banks.js
const xlsx = require('xlsx');
const normalize = (str) => String(str || '').trim().toLowerCase();

const bankConfigs = [
  {
    id: 'BANCO_5',
    identifier: (workbook) => {
      const SheetNames = (workbook.SheetNames || []).map(normalize);
      return SheetNames.includes('fixed income & cash') && SheetNames.includes('equity');
    },
    sheets: [
      {
        sheetName: 'Fixed Income & Cash',
        symbolColumn: 'Description',
        valueColumn: 'Market Value (Position Currency)',
        headerRow: 2,
        skipFooterRows: 1,
        customEndRowIndex: 25
      },
      {
        sheetName: 'Equity',
        symbolColumn: 'Description',
        valueColumn: 'Market Value (Position Currency)',
        headerRow: 2,
        skipFooterRows: 1,
        customEndRowIndex: 14
      }
    ]
  },
  {
    id: 'BANCO_2',
    identifier: (workbook) => {
      try {
        const sheet = workbook.Sheets[workbook.SheetNames[0]];
        const data = xlsx.utils.sheet_to_json(sheet, { header: 1 });

        for(let i= 0; i < 20; i++) {
          const row = data[i];
          if(!Array.isArray(row)) continue;

          const found = row.some(cell => {
          if (typeof cell !== 'string') return false;
            // unifica saltos de línea, tabs y espacios en uno solo
            const txt = cell
            .toLowerCase()
            .replace(/\s+/g, ' ')
            .trim();
            return txt.includes('valoración (usd)');
          });

          if (found) {
            console.log(`Cabecera encontrada en la fila ${i + 1}`);
            return true;
          }
        }

        return false;

      } catch (err) {
```

```
      console.warn("Error:", err.message);
      return false;
    }
  },
  sheets: [
    {
      sheetName: 'Data',
      isComplexReport: true,
      valueColumnNameToFind: 'Valoración (USD)',
      symbolColumnIndex: 6
    }
  ]
},
{
  // BANCO 4: CSV con celdas basura que hay que filtrar.
  id: 'BANCO_4',
  identifier: (workbook) => {
    try {
      const firstSheet = workbook.Sheets[workbook.SheetNames[0]];
      const headers = (xlsx.utils.sheet_to_json(firstSheet, { header: 1 })[0] || []).map(normalize);
      return headers.includes('description 1') && headers.includes('number/amt.');
    } catch { return false; }
  },
  sheets: [
    {
      sheetName: 0,
      headerRow: 1,
      symbolColumn: 'Description 1',
      valueColumn: 'Number/Amt.',
      rowFilter: (row) => row && row[0] && normalize(row[0]) !== 'description 1'
    }
  ]
},
{
  // BANCO 3: CSV estándar.
  id: 'BANCO_3',
  identifier: (workbook) => {
    try {
      const firstSheet = workbook.Sheets[workbook.SheetNames[0]];
      const headers = (xlsx.utils.sheet_to_json(firstSheet, { header: 1 })[0] || []).map(normalize);
      return headers.includes('description') && headers.includes('current value (reporting ccy)');
    } catch { return false; }
  },
  sheets: [
    {
      sheetName: 0,
      symbolColumn: 'Description',
      valueColumn: 'Current Value (Reporting CCY)'
    }
  ]
},
{
  // BANCO 1: CSV estándar.
  id: 'BANCO_1',
  identifier: (workbook) => {
    try {
      const firstSheet = workbook.Sheets[workbook.SheetNames[0]];
      const headers = (xlsx.utils.sheet_to_json(firstSheet, { header: 1 })[0] || []).map(normalize);
      return headers.includes('description') && headers.includes('value');
    } catch { return false; }
  },
  sheets: [
```

```
    {
      sheetName: 0,
      symbolColumn: 'Description',
      valueColumn: 'Value'
    }
  ]
 },
 {
  //Nuevos bancos con características nuevas
  id: 'BANCO_NUEVO',
  identifier: (workbook) => {
   try {
    const firstSheet = workbook.Sheets[workbook.SheetNames[0]];
    const headers = (xlsx.utils.sheet_to_json(firstSheet, { header: 1})[0] || []).map(normalize);
    const hasDescription = headers.some(h => ['description', 'descripcion', 'nombre', 'categoria'].includes(h));
    const hasValue = headers.some(h => ['value', 'valor', 'monto', 'market value', 'position', 'valoracion'].includes(h));

    if(hasDescription && hasValue) {
     console.log(`[BANCO_NUEVO] Cabecera reconocida. Usando una nueva configuración`);
     return true;
    }
    return false;
    } catch (err) {
     console.warn(`[BANCO_NUEVO] Error al identificar:`, err.message);
     return false;
   }
  },
  sheets: [
   {
    sheetName: 0,
    headerRow: 1,
    symbolColumn: (header) => {
     const normalized = header.map(normalize);
     return normalized.findIndex(h =>['decription','descripcion', 'nombre', 'categoria'].includes(h));
    },
    valueColumn: (header) => {
     const normalized = header.map(normalize);
     return normalized.findIndex(h => ['value', 'valor', 'monto', 'market value', 'position'].includes(h));
    }
   }
  ]
 }
];

module.exports = { bankConfigs };
```

Anexo 3. Parser.js

```
// src/main/parser.js
const xlsx = require('xlsx');
const path = require('path');
const {sugerenciasColumnas} = require('./fileUtils');

const normalize = (str) => String(str || '').replace(/\s+/g, ' ').trim().toLowerCase();

function identifyBank(fileBuffer) {
 try {
   const workbook = xlsx.read(fileBuffer, { type: 'buffer' });
   const { bankConfigs } = require('./banks/banks');
   return bankConfigs.find(config => config.identifier(workbook)) || null;
 } catch (error) {
   console.error("Error al leer el archivo para identificación:", error);
   sugerenciasColumnas();
   return null;
 }
}

function parseWithConfig(fileBuffer, config, filePath) {
   try {
      const workbook = xlsx.read(fileBuffer, { type: 'buffer' });
      let allAssets = [];

      for (const sheetConfig of config.sheets) {
         const sheetName = typeof sheetConfig.sheetName === 'number' ?
workbook.SheetNames[sheetConfig.sheetName] : sheetConfig.sheetName;
         if (!workbook.Sheets[sheetName]) continue;

         const worksheet = workbook.Sheets[sheetName];

         if (sheetConfig.isComplexReport) {
            allAssets.push(..._parseComplexReport(worksheet, sheetConfig, filePath));
         } else {
            allAssets.push(..._parseStandardTable(worksheet, sheetConfig, filePath));
         }
      }
      return allAssets;
   } catch (error) {
      console.error(`[Parser] Error fatal al procesar ${path.basename(filePath)}:`, error);
      return [];
   }
}

function _parseStandardTable(worksheet, sheetConfig, filePath) {
   const data = xlsx.utils.sheet_to_json(worksheet, { header: 1, defval: null });
   if (data.length === 0) return [];

   const allAssets = [];
   const headerRowIndex = (sheetConfig.headerRow || 1) - 1;
   const rawHeader = data[headerRowIndex] || [];
   const normalizedHeader = rawHeader.map(normalize);

   const symbolColIndex = typeof sheetConfig.symbolColumn === 'string' ? normalizedHeader.findIndex(h => h ===
normalize(sheetConfig.symbolColumn)) : sheetConfig.symbolColumn;
   const valueColIndex = typeof sheetConfig.valueColumn === 'string' ? normalizedHeader.findIndex(h => h ===
normalize(sheetConfig.valueColumn)) : sheetConfig.valueColumn;

   if (symbolColIndex === -1 || valueColIndex === -1) {
```

```
        console.warn(`[Parser Standard] No se encontraron las columnas requeridas en la hoja
'${sheetConfig.sheetName}'.`);
        return [];
    }

    const effectiveStartIndex = headerRowIndex + 1;
    const endRowIndex = data.length - (sheetConfig.skipFooterRows || 0);
    for (let i = effectiveStartIndex; i < endRowIndex; i++) {
        const row = data[i];
        if (!row || (sheetConfig.rowFilter && !sheetConfig.rowFilter(row))) continue;

        const symbol = row[symbolColIndex];
        const rawValue = row[valueColIndex];
        let numericValue = NaN;

        if (rawValue !== null && rawValue !== undefined) {
            numericValue = typeof rawValue === 'string' ? parseFloat(rawValue.replace(/,/g, '')) : parseFloat(rawValue);
        }

        if (symbol && !isNaN(numericValue) && numericValue !== 0) {
            allAssets.push({
                rawDescription: String(symbol).trim(),
                marketValue: numericValue,
                sourceFile: path.basename(filePath)
            });
        }
    }
    return allAssets;
}


function _parseComplexReport(worksheet, sheetConfig, filePath) {
    const allAssets = [];
    const data = xlsx.utils.sheet_to_json(worksheet, { header: 1, defval: null });

    let headerRowIndex = -1;
    let valueColIndex = -1;
    const symbolColIndex = sheetConfig.symbolColumnIndex ?? 6;

    for (let i = 0; i < 20 && i < data.length; i++) {
        const row = data[i];
        if(!Array.isArray(row)) continue;
        const index = row.findIndex(cell => normalize(cell) === normalize(sheetConfig.valueColumnNameToFind));
        if(index !== -1) {
            valueColIndex = index;
            headerRowIndex = i;
            break;
        }
    }

    if(valueColIndex === -1) {
        console.error("No se encontró la columna 'Valoración económica'");
        return [];
    }

    const startRowIndex = headerRowIndex +1;

    for (let R = startRowIndex; R < data.length; R++) {
        const row = data[R];
        if (!Array.isArray(row)) continue;

        const description = row[symbolColIndex];
```

```
        const value = row[valueColIndex];

        const validDescription = typeof description === 'string' && description.trim() !== '';
        const validValue = typeof value === 'number' && !isNaN(value) && value !== 0;

        if (validDescription && validValue) {
            allAssets.push({
                rawDescription: description.trim(),
                marketValue: value,
                sourceFile: path.basename(filePath)
            });
        }
    }
    console.log(`[Parser ComplexReport] Se encontraron ${allAssets.length} activos.`);
    return allAssets;
}

module.exports = { parseWithConfig, identifyBank };
```

Anexo 4. Rebalancer.js

```
//src/main/rebalancer.js

console.log("--- CARGANDO: ./rebalancer.js ---");

function generateFullPortfolioSummary(portfolio, riskProfileData, riskProfileColumn, categoriesData) {
    const warnings = [];

    function cleanText(str) {
        return String(str || '')
            .toLowerCase()
            .trim();
    }
    // Mapa subcategoría → categoría
    const subcatToCatMap = {};
    for (const catRow of categoriesData) {
        if (catRow['Subcategoria'] && catRow['Categoria']) {
            subcatToCatMap[cleanText(catRow['Subcategoria'])] = cleanText(catRow['Categoria']);
        }
    }

    // Mapa de perfil: clave = cat|subcat
    const riskProfileDict = {};
    const categoryTotals = {};
    const allKeys = Object.keys(riskProfileData[0]);
    const labelColumnName = allKeys.find(key => key.trim().toLowerCase().includes('risk profile:'));
    if (!labelColumnName) {
        throw new Error(`No se pudo encontrar la columna "Risk Profile" en el archivo de perfiles. Columnas encontradas:
${allKeys.join(', ')}`);
    }

    let currentSuperCategory = null;
    let currentyCategory = null;

    for(let i = 0; i < riskProfileData.length; i++){
        const row = riskProfileData[i];
        const label = row[labelColumnName];
        const value = row[riskProfileColumn];
        const labelAsString = String(label || '').trim();
```

```
    if(!labelAsString || labelAsString.toUpperCase() === 'TOTAL:') continue;

    const isAllCaps = labelAsString === labelAsString.toUpperCase();

    let numericValue = 0;
    if(typeof value === 'string'){
       numericValue = parseFloat(value.replace(',', '.').replace('%', '').trim()) / 100;
    } else if (typeof value === 'number') {
       numericValue = value >= 1 ? value / 100 : value;
    }

    if(isAllCaps) {
       currentSuperCategory = cleanText(labelAsString);
       currentyCategory = null;
       let hasSubcategory = false;
       for (let j = i + 1; j < riskProfileData.length; j++){
          const nextLabel = String(riskProfileData[j]?.[labelColumnName] || '').trim();
          if(!nextLabel || nextLabel.toUpperCase() === 'TOTAL:') break;
          if(nextLabel === nextLabel.toUpperCase()) break;
          hasSubcategory = true;
          break;
       }

       if(!hasSubcategory && !isNaN(numericValue) && numericValue > 0) {
          const key = `${currentSuperCategory}|${currentSuperCategory}`;
          riskProfileDict[key] = numericValue;
          categoryTotals[currentSuperCategory] = (categoryTotals[currentSuperCategory] || 0) + numericValue;
       }
       continue;
    }

    const cleanedLabel = cleanText(labelAsString);

    let hasSubSub = false;
    for(let j = i + 1; j < riskProfileData.length; j++){
       const nextLabel = String(riskProfileData[j]?.[labelColumnName] || '').trim();
       if(!nextLabel || nextLabel.toUpperCase() === 'TOTAL:') break;
       const nextValue = riskProfileData[j]?.[riskProfileColumn];
       if(typeof nextValue === 'string' || typeof nextValue === 'number') {
          hasSubSub = true;
          break;
       }
    }

    if(hasSubSub) {
       currentyCategory = cleanedLabel;
       if(!isNaN(numericValue)) {
          const key = `${currentSuperCategory}|${currentyCategory}`;
          riskProfileDict[key] = numericValue;
          categoryTotals[currentSuperCategory] = (categoryTotals[currentSuperCategory] || 0) + numericValue;
       }
       continue;
    }

    const cat = currentyCategory || currentSuperCategory || 'n/a';
    const key = cleanedLabel === cat ? `${cat}|${cat}` : `${cat}|${cleanedLabel}`;

    if(!isNaN(numericValue)) {
       riskProfileDict[key] = numericValue;
       categoryTotals[cat] = (categoryTotals[cat] || 0) + numericValue;
    }
 }
```

```
if (Object.keys(riskProfileDict).length === 0) {
    throw new Error(`No se encontraron datos de subcategorías válidos para el perfil '${riskProfileColumn}'.`);
}

const totalMarketValue = portfolio.reduce((sum, item) => sum + (item['Valor de Mercado'] || 0), 0);

const grouped = {};
for (const item of portfolio) {
    if (item.Subcategoria === 'NO RECONOCIDOS') continue;

    const cat = cleanText(item.Categoria);
    const subcat = cleanText(item.Subcategoria);
    const key = `${cat}|${subcat}`;

    if (!grouped[key]) {
        grouped[key] = {
            Categoria: item.Categoria,
            Subcategoria: item.Subcategoria,
            'Valor en Portafolio': 0
        };
    }
    grouped[key]['Valor en Portafolio'] += item['Valor de Mercado'];
}

const summary = [];
const processedKeys = new Set();

// 1. Procesar claves presentes en el perfil
for (const key in riskProfileDict) {
    processedKeys.add(key);
    const [cat, subcat] = key.split('|');

    const group = grouped[key] || {
        Categoria: cat.toUpperCase(),
        Subcategoria: subcat.toUpperCase(),
        'Valor en Portafolio': 0
    };

    const profilePercentage = riskProfileDict[key];
    const valorPerfil = profilePercentage * totalMarketValue;
    const diferencia = group['Valor en Portafolio'] - valorPerfil;
    const accion = diferencia.toFixed(2) == 0.00 ? 'Mantener' : (diferencia < 0 ? 'Compra' : 'Venta');

    summary.push({
        Categoria: group.Categoria,
        Subcategoria: group.Subcategoria,
        'Valor en Portafolio': group['Valor en Portafolio'],
        'Porcentaje en Portafolio': totalMarketValue > 0 ? (group['Valor en Portafolio'] / totalMarketValue) : 0,
        'Porcentaje de Perfil': profilePercentage,
        'Valor de Perfil': valorPerfil,
        'Diferencia $': diferencia,
        'Acción': accion
    });
}

// 2. Detectar subcategorías del portafolio no presentes en el perfil
for (const key in grouped) {
    if (processedKeys.has(key)) continue;
    const group = grouped[key];
    summary.push({
        Categoria: group.Categoria,
```

```
            Subcategoria: group.Subcategoria,
            'Valor en Portafolio': group['Valor en Portafolio'],
            'Porcentaje en Portafolio': totalMarketValue > 0 ? (group['Valor en Portafolio'] / totalMarketValue) : 0,
            'Porcentaje de Perfil': 0,
            'Valor de Perfil': 0,
            'Diferencia $': -group['Valor en Portafolio'],
            'Acción': 'Venta'
        });
    }

    return { summary, total: totalMarketValue, warnings };
}

module.exports = { generateFullPortfolioSummary };
```

Anexo 5. Renderer.js

```
// src/renderer/renderer.js

import { groupDataForCharts, getColorFor } from '../main/UtilsCharts.js';

let lastSummaryData = [];

// ----------------------------------------
// Setup de inputs de archivos
// ----------------------------------------
function setupFileInput(inputId, labelId) {
  const fileInput = document.getElementById(inputId);
  const infoLabel = document.getElementById(labelId);
  if (!fileInput || !infoLabel) {
    console.warn(`Elementos no encontrados para el input: ${inputId}`);
    return;
  }
  fileInput.addEventListener('change', (e) => {
    const files = e.target.files;
    infoLabel.textContent = files.length === 0
      ? 'Ningún archivo seleccionado'
      : fileInput.multiple
        ? `${files.length} archivo(s) seleccionado(s)`
        : files[0].name;
  });
}

// ----------------------------------------
// Inicialización UI y listeners
// ----------------------------------------
function initializeUI() {
  setupFileInput('bankFile', 'bankFilesLabel');
  setupFileInput('categoriesFile', 'categoriesFileLabel');
  setupFileInput('riskProfileFile', 'profileFileLabel');
  setupFileInput('comparisonFile', 'compareFilesLabel');

  document.getElementById('process-button')
    .addEventListener('click', handleProcessButtonClick);
  document.getElementById('refresh-button')
    .addEventListener('click', () => window.location.reload());
  document.getElementById('logout-button')
    .addEventListener('click', () => {window.sessionStorage.removeItem('logged-In');
      window.location.reload();});
```

```
  document.getElementById('close-button')
    .addEventListener('click', () => window.electronAPI.closeApp());
  document.getElementById('toggle-sidebar')
    .addEventListener('click', () =>
      document.getElementById('sidebar').classList.toggle('collapsed')
    );
  document.querySelectorAll('#tabs .stTab').forEach(tab => {
    tab.addEventListener('click', () => {
      document
        .querySelectorAll('#tabs .stTab')
        .forEach(t => t.classList.remove('active'));
      tab.classList.add('active');
      document
        .querySelectorAll('.tab-section')
        .forEach(s => s.classList.add('hidden'));
      const sec = document.getElementById(tab.dataset.tab);
      if (sec) {
        sec.classList.remove('hidden');
        if (tab.dataset.tab === 'charts') updateChartsSection();
      }
    });
  });

  const select = document.getElementById('toggle-chart-type');
  if (select) {
    select.addEventListener('change', () => {
      if (lastSummaryData.length) renderCharts(lastSummaryData);
    });
  }

// Ocultar el header de gráficos al inicializar
  hideChartHeader();
}

// Ocultar header de gráficos
function hideChartHeader() {
  const chartHeader = document.querySelector('.chart-header');
  if (chartHeader) {
    chartHeader.style.display = 'none';
  }
}

// Mostrar header de gráficos
function showChartHeader() {
  const chartHeader = document.querySelector('.chart-header');
  if (chartHeader) {
    chartHeader.style.display = 'flex';
  }
}

// ----------------------------------------
// Función para mostrar banner de advertencias
// ----------------------------------------
function showWarningsBanner(warningsCount) {
  // Remover banners existentes
  document.querySelectorAll('.warnings-banner').forEach(banner => banner.remove());

  if (warningsCount > 0) {
    const targetTabs = ['summary', 'charts', 'comparison'];

    targetTabs.forEach(tabId => {
      const tabSection = document.getElementById(tabId);
```

```
    if (tabSection) {
      const warningBanner = document.createElement('div');
      warningBanner.className = 'warnings-banner';
      warningBanner.style.cssText = `
        background: linear-gradient(135deg, #ff6b6b, #ee5a52);
        color: white;
        padding: 12px 20px;
        margin: 0 0 20px 0;
        border-radius: 8px;
        font-weight: 600;
        text-align: center;
        box-shadow: 0 3px 10px rgba(255, 107, 107, 0.3);
        border-left: 5px solid #dc3545;
        display: flex;
        align-items: center;
        justify-content: center;
        gap: 10px;
        cursor: pointer;
        transition: all 0.3s ease;
      `;

      warningBanner.innerHTML = `
        <span style="font-size: 1.2em;">⚠</span>
        <span>Se detectaron <strong>${warningsCount}</strong> advertencia${warningsCount > 1 ? 's' : ''}!
        Por favor ve a la pestaña de <strong>Advertencias</strong> y categoriza los activos faltantes.</span>
      `;

      // Agregar efecto hover y click
      warningBanner.addEventListener('mouseover', () => {
        warningBanner.style.transform = 'translateY(-2px)';
        warningBanner.style.boxShadow = '0 5px 15px rgba(255, 107, 107, 0.4)';
      });

      warningBanner.addEventListener('mouseout', () => {
        warningBanner.style.transform = 'translateY(0)';
        warningBanner.style.boxShadow = '0 3px 10px rgba(255, 107, 107, 0.3)';
      });

      // Click para ir a la pestaña de advertencias
      warningBanner.addEventListener('click', () => {
        document.querySelectorAll('#tabs .stTab').forEach(t => t.classList.remove('active'));
        document.querySelectorAll('.tab-section').forEach(s => s.classList.add('hidden'));

        const warningsTab = document.querySelector('[data-tab="warnings"]');
        const warningsSection = document.getElementById('warnings');

        if (warningsTab && warningsSection) {
          warningsTab.classList.add('active');
          warningsSection.classList.remove('hidden');
        }
      });

      // Insertar al inicio de la sección
      tabSection.insertBefore(warningBanner, tabSection.firstChild);
    }
  });
 }
}

// ---------------------------------------
// Procesamiento principal
// ---------------------------------------
```

```
async function handleProcessButtonClick() {
  const btn = document.getElementById('process-button');
  const warn = document.getElementById('warning-banner');
  warn.style.display = 'none';
  btn.disabled = true;
  btn.textContent = 'Procesando...';

  const banks = Array.from(document.getElementById('bankFile').files);
  const cats  = document.getElementById('categoriesFile').files[0];
  const prof  = document.getElementById('riskProfileFile').files[0];
  const comp  = Array.from(document.getElementById('comparisonFile').files);
  const sel   = document.getElementById('riskProfileSelect').value.trim();

  if (!banks.length) return showWarning('Selecciona al menos un banco'), resetBtn();
  if (!cats)         return showWarning('Selecciona el archivo de categorías'), resetBtn();
  if (!prof)         return showWarning('Selecciona el perfil de riesgo'), resetBtn();
  if (!sel)          return showWarning('Selecciona un perfil del menú'), resetBtn();

  try {
    const payload = {
      bankFilePaths: Array.from(banks).map(f => f.path),
      categoriesFilePath: cats.path,
      profileFilePath: prof.path,
      compareFilePaths: Array.from(comp).map(f => f.path),
      selectedProfile: sel
    };
    const { success, data, error } = await window.electronAPI.runFullAnalysis(payload);
    if (!success) throw new Error(error);

    const { summary, total, warnings, comparisonData, uncategorizedCapital } = data;
    if (!summary?.length) throw new Error('Resumen vacío');

    renderSummaryTable(summary, total, uncategorizedCapital);
    renderCharts(summary);
    renderWarnings(warnings);
    updateTabBadges(summary, warnings);
    if (comparisonData?.length) renderComparisonTable(comparisonData);

    showWarningsBanner(warnings.length);

    showChartHeader();

    lastSummaryData = summary;
  } catch (e) {
    console.error(e);
    showWarning(`Error: ${e.message}`);
  } finally {
    resetBtn();
  }
}

function resetBtn() {
  const btn = document.getElementById('process-button');
  btn.disabled = false;
  btn.textContent = 'Procesar';
}

function showWarning(msg) {
  const warn = document.getElementById('warning-banner');
  warn.textContent = msg;
  warn.style.display = 'block';
  window.scrollTo(0, 0);
```

```
}

function updateTabBadges(summary, warnings) {
  const safe = sel => document.querySelector(sel) || { textContent: '' };
  safe('#summary-tab-btn .badge').textContent  = summary.length;
  safe('#charts-tab-btn .badge').textContent   = '4';
  safe('#warnings-tab-btn .badge').textContent = warnings.length;
}

function renderWarnings(warnings) {
  const sec = document.getElementById('warnings');
  sec.innerHTML = '';

  if (warnings?.length) {
    // Título principal
    const titleDiv = document.createElement('div');
    titleDiv.style.cssText = `
      margin-bottom: 20px;
      text-align: center;
    `;
    titleDiv.innerHTML = `<h3 style="color: #495057; font-weight: 600; margin: 0;">Se encontraron ${warnings.length}
advertencia${warnings.length > 1 ? 's' : ''}:</h3>`;
    sec.appendChild(titleDiv);

    // Contenedor para las advertencias
    const warningsContainer = document.createElement('div');
    warningsContainer.style.cssText = `
      display: flex;
      flex-direction: column;
      gap: 12px;
      max-height: 70vh;
      overflow-y: auto;
      padding: 10px 0;
    `;

    warnings.forEach((warning, index) => {
      const warningDiv = document.createElement('div');
      warningDiv.style.cssText = `
        background: linear-gradient(135deg, #ff6b6b, #ee5a52);
        color: white;
        padding: 16px 20px;
        border-radius: 8px;
        font-weight: 500;
        box-shadow: 0 3px 10px rgba(255, 107, 107, 0.3);
        border-left: 5px solid #dc3545;
        transition: all 0.3s ease;
        line-height: 1.4;
        font-size: 0.95rem;
      `;

      // Efectos hover
      warningDiv.addEventListener('mouseover', () => {
        warningDiv.style.transform = 'translateY(-2px)';
        warningDiv.style.boxShadow = '0 5px 15px rgba(255, 107, 107, 0.4)';
      });

      warningDiv.addEventListener('mouseout', () => {
        warningDiv.style.transform = 'translateY(0)';
        warningDiv.style.boxShadow = '0 3px 10px rgba(255, 107, 107, 0.3)';
      });

      warningDiv.textContent = warning;
```

```
      warningsContainer.appendChild(warningDiv);
    });

    sec.appendChild(warningsContainer);

    // Mensaje de ayuda al final
    const helpDiv = document.createElement('div');
    helpDiv.style.cssText = `
      margin-top: 20px;
      padding: 16px;
      background: linear-gradient(135deg, #17a2b8, #138496);
      color: white;
      border-radius: 8px;
      text-align: center;
      font-weight: 500;
      box-shadow: 0 3px 10px rgba(23, 162, 184, 0.3);
    `;
    helpDiv.innerHTML = `
      <strong>💡 Sugerencia:</strong> Para categorizar estos activos, agrega las descripciones faltantes al archivo de categorías
      o revisa si hay errores tipográficos en los nombres de los activos.
    `;
    sec.appendChild(helpDiv);

  } else {
    // Estado sin advertencias
    const noWarningsDiv = document.createElement('div');
    noWarningsDiv.style.cssText = `
      text-align: center;
      padding: 40px 20px;
      background: #2E7D32;
      color: white;
      border-radius: 12px;
      box-shadow: 0 4px 12px rgba(40, 167, 69, 0.3);
    `;
    noWarningsDiv.innerHTML = `
      <h3 style="color: white; margin: 0; font-size: 1.2rem; font-weight: 600;">Todos los activos fueron categorizados correctamente</h3>
    `;
    sec.appendChild(noWarningsDiv);
  }
}

// ----------------------------------------
// Renderiza la tabla de Resumen
// ----------------------------------------
function renderSummaryTable(data, total, uncategorizedCapital) {
  const sec = document.getElementById('summary');
  sec.innerHTML = '';

  const invisibles = ['RENTA FIJA GRADO DE INVERSIÓN', 'RENTA FIJA ALTO RENDIMIENTO'];
  const groupedFiltered = data.filter(row => !invisibles.includes((row['Subcategoria']|| '').toUpperCase()));

  if(!groupedFiltered.length) return;
  const cols = Object.keys(groupedFiltered[0]);

  console.log('Capital faltante final para mostrar:', uncategorizedCapital);

  // Crear contenedor principal para las métricas
  const metricsContainer = document.createElement('div');
  metricsContainer.style.cssText = `
    display: flex;
```

```
    gap: 20px;
    margin-bottom: 30px;
    flex-wrap: wrap;
  `;

  // Crear div para Total Portafolio
  const totalDiv = document.createElement('div');
  totalDiv.style.cssText = `
    background: #2E7D32;
    color: white;
    padding: 20px;
    border-radius: 12px;
    box-shadow: 0 4px 12px rgba(40, 167, 69, 0.3);
    flex: 1;
    min-width: 300px;
    text-align: center;
  `;

  totalDiv.innerHTML = `
    <h3 style="color: white; margin-top: 0; font-size: 1.2rem; font-weight: 600;">Total Portafolio</h3>
    <p style="font-size: 2.2rem; color: white; margin: 10px 0; font-weight: bold; text-shadow: 0 2px 4px rgba(0,0,0,0.2);">
      $${total.toLocaleString(undefined, { minimumFractionDigits: 2, maximumFractionDigits: 2 })}
    </p>
    <p style="color: rgba(255,255,255,0.9); margin-bottom: 0; font-size: 0.95rem;">
      Valor total de activos categorizados
    </p>
  `;

  // Crear div para Capital Faltante (solo si hay capital faltante)
  metricsContainer.appendChild(totalDiv);

  if (uncategorizedCapital > 0) {
    const warningDiv = document.createElement('div');
    warningDiv.style.cssText = `
      background: #F57C00;
      color: white;
      padding: 20px;
      border-radius: 12px;
      box-shadow: 0 4px 12px rgba(255, 193, 7, 0.3);
      flex: 1;
      min-width: 300px;
      text-align: center;
    `;

    const totalCompleto = total + uncategorizedCapital;
    const porcentajeFaltante = ((uncategorizedCapital / totalCompleto) * 100).toFixed(2);

    warningDiv.innerHTML = `
      <h3 style="color: white; margin-top: 0; font-size: 1.2rem; font-weight: 600;">⚠️ Capital Faltante de Categorizar</h3>
      <p style="font-size: 2.2rem; color: white; margin: 10px 0; font-weight: bold; text-shadow: 0 2px 4px
rgba(0,0,0,0.2);">
        $${uncategorizedCapital.toLocaleString(undefined, { minimumFractionDigits: 2, maximumFractionDigits: 2 })}
      </p>
      <p style="color: rgba(255,255,255,0.9); margin-bottom: 0; font-size: 0.95rem;">
        ${porcentajeFaltante}% del portafolio total (activos no categorizados)
      </p>
    `;

    metricsContainer.appendChild(warningDiv);
  }

  sec.appendChild(metricsContainer);
```

```
// Crear contenedor responsivo para la tabla
const tableContainer = document.createElement('div');
tableContainer.style.cssText = `
  width: 100%;
  overflow-x: auto;
  background: white;
  border-radius: 12px;
  box-shadow: 0 4px 12px rgba(0, 0, 0, 0.1);
  border: 1px solid #e9ecef;
`;

// Crear y agregar la tabla
const tbl = document.createElement('table');
tbl.style.cssText = `
  width: 100%;
  border-collapse: collapse;
  margin: 0;
  font-size: 0.9rem;
  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, sans-serif;
  background: white;
  min-width: 800px;
`;


// Crear header con diseño mejorado
let thead = '<thead><tr>';
cols.forEach(col => {
  thead += `<th style="
    background:  #0056b3;
    color: white;
    padding: 16px 12px;
    text-align: center;
    font-weight: 600;
    font-size: 0.85rem;
    text-transform: uppercase;
    letter-spacing: 0.5px;
    border: none;
    position: sticky;
    top: 0;
    z-index: 10;
    box-shadow: 0 2px 4px rgba(0,0,0,0.1);
  ">${col}</th>`;
});
thead += '</tr></thead>';

// Crear body con filas alternadas
let tbody = '<tbody>';
groupedFiltered.forEach((row, index) => {
  const isEven = index % 2 === 0;
  tbody += `<tr style="
    background-color: ${isEven ? '#ffffff' : '#f8f9fa'};
    transition: all 0.2s ease;
  " onmouseover="this.style.backgroundColor='#e3f2fd'; this.style.transform='scale(1.01)'"
     onmouseout="this.style.backgroundColor='${isEven ? '#ffffff' : '#f8f9fa'}'; this.style.transform='scale(1)'">`;

  cols.forEach(col => {
    let v = row[col];
    let cellStyle = 'padding: 12px; text-align: center; border: none; vertical-align: middle;';

    if (typeof v === 'number') {
      if (col.toLowerCase().includes('porcentaje')) {
```

```
        v = `${(v*100).toLocaleString(undefined,{minimumFractionDigits:2, maximumFractionDigits: 2})}%`;
        cellStyle += ' font-weight: 600; color: #495057;';
       } else {
        v = `${v.toLocaleString(undefined,{minimumFractionDigits:2, maximumFractionDigits: 2})}`;
        cellStyle += ' font-weight: 600; color: #28a745;';
       }
      } else if (col === 'Acción') {
       const actionColors = {
        'Compra': '#28a745',
        'Venta': '#dc3545',
        'Mantener': '#6c757d'
       };
       const bgColors = {
        'Compra': '#d4edda',
        'Venta': '#f8d7da',
        'Mantener': '#e2e3e5'
       };
       cellStyle += `color: ${actionColors[v] || '#495057'}; background-color: ${bgColors[v] || 'transparent'}; font-weight:
600; border-radius: 6px; margin: 2px;`;
      }
      if(col === 'Subcategoria' && typeof v === 'string'){
       v = v.toLowerCase().replace(/^\w/, c => c.toUpperCase());
      }

      tbody += `<td style="${cellStyle}">${v}</td>`;
    });
    tbody += '</tr>';
  });
  tbody += '</tbody>';

  tbl.innerHTML = thead + tbody;
  tableContainer.appendChild(tbl);
  sec.appendChild(tableContainer);
}

// ----------------------------------------
// SECTION: GRÁFICAS
// ----------------------------------------
function updateChartsSection() {
  if (!lastSummaryData.length) return;
  renderCharts(lastSummaryData);
}

function renderCharts(data) {
  const container = document.getElementById('charts-container');
  container.innerHTML = '';
  container.style.overflowX = 'hidden';

  const viewType = document.getElementById('toggle-chart-type').value;
  const { grouped, groupCol } = groupDataForCharts(data, viewType);
  if (!grouped.length) {
    container.innerHTML = '<p>No hay datos para mostrar.</p>';
    return;
  }

  const invisibles = ['RENTA FIJA GRADO DE INVERSIÓN', 'RENTA FIJA ALTO RENDIMIENTO'];
  const groupedFiltered = grouped.filter(row => !invisibles.includes(row[groupCol]?.toUpperCase()));

  // Contenedor principal
  const mainContainer = document.createElement('div');
  mainContainer.style.cssText = `
    max-width: 1400px;
```

```
  margin: 0 auto;
  padding: 0 20px;
  width: 100%;
  box-sizing: border-box;
`;

// — Filas —
const row1 = document.createElement('div');
const row2 = document.createElement('div');
const row3 = document.createElement('div');
[row1, row2, row3].forEach(r => {
  r.style.width = '100%';
  r.style.boxSizing = 'border-box';
});

// Alturas y separaciones
row1.style.cssText = `
  height: 480px;
  margin-bottom: 40px;
  width: 100%;
  flex-wrap: wrap;
`;
row2.style.cssText = `
  display: flex;
  gap: 20px;
  height: 450px;
  margin-bottom: 60px;
  width: 100%;
  flex-wrap: wrap;
  max-width: 1200px;
  margin-left: auto;
  margin-right: auto;
`;
row3.style.cssText = `
  display: flex;
  gap: 20px;
  width: 100%;
  felx-wrap: wrap;
  max-width: 1000px,
  margin-left: auto;
  margin-right: auto;
`;

// subdividir row2: 1/3 y 2/3
const opCol  = document.createElement('div');
const barCol = document.createElement('div');
opCol.style.flex  = '1';
opCol.style.minWidth = '300px';
barCol.style.flex    = '2';
barCol.style.minWidth = '400px';
row2.append(opCol, barCol);

// subdividir row3: 50/50
const pieA = document.createElement('div');
const pieB = document.createElement('div');
pieA.style.flex = '1';
pieB.style.flex = '1';
pieA.style.minWidth = '300px';
pieB.style.minWidth = '300px';
row3.append(pieA, pieB);

mainContainer.append(row1, row2, row3);
```

```
  container.appendChild(mainContainer);

  // 1) barras verticales
  createBarChart(groupedFiltered, groupCol, row1);

  // 2) operaciones + barras horizontales
  createMetricsAndHorizontalChart(groupedFiltered, groupCol, opCol, barCol);

  // 3) dos pies
  createPieCharts(groupedFiltered, groupCol, pieA, pieB);
}

function createBarChart(groupedFiltered, groupCol, container) {
  const div = document.createElement('div');
  div.style.height = '100%';
  div.style.width = '100%';
  container.appendChild(div);

  const t1 = {
    x: groupedFiltered.map(r => r[groupCol]),
    y: groupedFiltered.map(r => r['Valor en Portafolio']),
    name: 'Actual', type: 'bar', marker: { color: '#374c80' }
  };
  const t2 = {
    x: groupedFiltered.map(r => r[groupCol]),
    y: groupedFiltered.map(r => r['Valor de Perfil']),
    name: 'Perfil', type: 'bar', marker: { color: '#7a5195' }
  };

  // Configuración responsive con proporciones mejoradas
  const layout = {
    autosize: true,
    responsive: true,
    title: { text: 'Comparación de Valores de Mercado', font: { size: 16 } },
    barmode: 'group',
    margin: { b: 120, l: 80, r: 40, t: 60 },
    xaxis: {
      tickangle: -45,
      automargin: true,
      tickfont: { size: 12 }
    },
    yaxis: {
      tickformat: '$,.2f',
      automargin: true,
      tickfont: { size: 12 }
    },
    legend: {
      orientation: 'h',
      x: 0.5,
      xanchor: 'left',
      y: 1.1,
      font: { size: 10 },
      itemwidth: 100,
      valign: 'middle'
    }
  };

  const config = {
    responsive: true,
    displayModeBar: false
  };
```

```
  Plotly.newPlot(div, [t1, t2], layout, config);
}

function createMetricsAndHorizontalChart(groupedFiltered, groupCol, opsContainer, chartContainer) {
  groupedFiltered.forEach(row => {
    row.Delta = row['Valor de Perfil'] - row['Valor en Portafolio'];
  });
  const compras = groupedFiltered.filter(d => d.Delta > 0);
  const ventas  = groupedFiltered.filter(d => d.Delta < 0);

  // tabla operaciones
  const md = document.createElement('div');
  md.style.padding      = '20px';
  md.style.background   = '#f8f9fa';
  md.style.borderRadius = '8px';
  md.style.overflowY    = 'auto';
  md.style.height       = '100%';
  md.innerHTML = '<h3>Operaciones Requeridas</h3>';
  if (compras.length) {
    md.innerHTML += '<h4 style="color: #28a745;">📈 Compras</h4>';
    compras.forEach(i => {
      md.innerHTML += `<div style="padding:8px;border-left:4px solid #28a745;margin-bottom:8px;">
                         <strong>${i[groupCol]}:</strong> ${Math.abs(i.Delta).toLocaleString()}
                       </div>`;
    });
  }
  if (ventas.length) {
    md.innerHTML += '<h4 style="color: #dc3545;">📉 Ventas</h4>';
    ventas.forEach(i => {
      md.innerHTML += `<div style="padding:8px;border-left:4px solid #dc3545;margin-bottom:8px;">
                         <strong>${i[groupCol]}:</strong> ${Math.abs(i.Delta).toLocaleString()}
                       </div>`;
    });
  }
  if (!compras.length && !ventas.length) {
    md.innerHTML += '<p style="font-style:italic;color:#6c757d;">No se requieren operaciones.</p>';
  }
  opsContainer.appendChild(md);

  // gráfica horizontal
  const cd = document.createElement('div');
  cd.style.height = '100%';
  cd.style.width = '100%';
  chartContainer.appendChild(cd);

  const trace = {
    x: groupedFiltered.map(d => d.Delta),
    y: groupedFiltered.map(d => d[groupCol]),
    type: 'bar',
    orientation: 'h',
    marker: {
      color: groupedFiltered.map(d => d.Delta > 0 ? '#28a745' : '#dc3545')
    }
  };

  const layout = {
    autosize: true,
    responsive: true,
    title: { text: 'Operaciones por Categoría', font: { size: 16 } },
    margin: { l: 150, r: 40, t: 60, b: 60 },
    xaxis: {
      title: 'Monto ($)',
```

```
      automargin: true,
      tickfont: { size: 11 }
    },
    yaxis: {
      automargin: true,
      tickfont: { size: 11 }
    }
  };

  const config = {
    responsive: true,
    displayModeBar: false
  };

  Plotly.newPlot(cd, [trace], layout, config);
}

function createPieCharts(groupedFiltered, groupCol, pieA, pieB) {
  const labels = groupedFiltered.map(d => d[groupCol]);
  const valsA  = groupedFiltered.map(d => d['Valor en Portafolio'] === 0 ? 1e-6 : d['Valor en Portafolio']);
  const valsP  = groupedFiltered.map(d => d['Valor de Perfil'] === 0 ? 1e-6 : d['Valor de Perfil']);

  [pieA, pieB].forEach(c => c.style.height = '100%');

  // pie actual (solo porcentaje)
  Plotly.newPlot(pieA, [{
    labels, values: valsA, type: 'pie', hole: 0.4,
    textinfo: 'percent+label',
    textposition: 'outside',
    marker: {
      colors: groupedFiltered.map(d => getColorFor(d[groupCol]) || '#cccccc'),
      line: { color: '#ffffff', width: 2 }
    }
  }], {
    autosize: true,
    height: null,
    title: { text: 'Proporción de Inversión Actual', font: { size: 16 } },
    legend: {
      orientation: 'h',
      x: 0.5,
      xanchor: 'auto',
      y: -0.5,
      font: {size: 11},
      itemwidth: 100
    },
    margin: { t: 60, b: 80, l: 40, r: 40 }
  });

  // pie objetivo (solo porcentaje)
  Plotly.newPlot(pieB, [{
    labels, values: valsP, type: 'pie', hole: 0.4,
    textinfo: 'percent+label',
    textposition: 'outside',
    marker: {
      colors: groupedFiltered.map(d => getColorFor(d[groupCol]) || '#cccccc'),
      line: { color: '#ffffff', width: 2 }
    }
  }], {
    autosize: true,
    height: null,
    title: { text: 'Proporción de Perfil Objetivo', font: { size: 16 } },
    legend: {
```

```
      orientation: 'h',
      x: 0.5,
      xanchor: 'auto',
      y: -0.3,
      font: {size: 11},
      itemwidth: 100
    },
    margin: { t: 60, b: 80, l: 40, r: 40 }
  });
}

// ----------------------------------------
// Renderiza la tabla de Comparación
// ----------------------------------------
function renderComparisonTable(comparisonData) {
  const c = document.getElementById('comparison');
  if (!c) return;

  c.innerHTML = `
    <h3 style="margin-bottom: 20px; color: #000000; font-weight: 600;">Análisis Comparativo</h3>
  `;

  if (!comparisonData.length) {
    c.innerHTML += '<p style="text-align: center; color: #6c757d; font-style: italic;">No hay datos comparables.</p>';
    return;
  }

  const invisibles = ['RENTA FIJA GRADO DE INVERSIÓN', 'RENTA FIJA ALTO RENDIMIENTO'];
  const groupedFiltered = comparisonData.filter(row => !invisibles.includes((row['Subcategoria']|| '').toUpperCase()));

  if(!groupedFiltered.length) return;
  const cols = Object.keys(groupedFiltered[0]);

  // Crear contenedor responsivo para la tabla
  const tableContainer = document.createElement('div');
  tableContainer.style.cssText = `
    width: 100%;
    overflow-x: auto;
    background: white;
    border-radius: 12px;
    box-shadow: 0 4px 12px rgba(0, 0, 0, 0.1);
    border: 1px solid #e9ecef;
  `;

  const tbl = document.createElement('table');
  tbl.style.cssText = `
    width: 100%;
    border-collapse: collapse;
    margin: 0;
    font-size: 0.9rem;
    font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, sans-serif;
    background: white;
    min-width: 700px;
  `;


  // Crear header con el mismo estilo que la tabla de resumen
  let thead = '<thead><tr>';
  cols.forEach(col => {
    thead += `<th style="
      background:  #0056b3;
      color: white;
```

```
    padding: 16px 12px;
    text-align: center;
    font-weight: 600;
    font-size: 0.85rem;
    text-transform: uppercase;
    letter-spacing: 0.5px;
    border: none;
    position: sticky;
    top: 0;
    z-index: 10;
    box-shadow: 0 2px 4px rgba(0,0,0,0.1);
  ">${col}</th>`;
});
thead += '</tr></thead>';

// Crear body con filas alternadas y estilos especiales
let tbody = '<tbody>';
groupedFiltered.forEach((row, index) => {
  const isEven = index % 2 === 0;
  tbody += `<tr style="
    background-color: ${isEven ? '#ffffff' : '#f8f9fa'};
    transition: all 0.2s ease;
  " onmouseover="this.style.backgroundColor='#e3f2fd'; this.style.transform='scale(1.01)'"
     onmouseout="this.style.backgroundColor='${isEven ? '#ffffff' : '#f8f9fa'}'; this.style.transform='scale(1)'">`;

  cols.forEach(col => {
    let v = row[col];
    let cellStyle = 'padding: 12px; text-align: center; border: none; vertical-align: middle;';

    if (typeof v === 'number') {
      if (col.includes('($)')) {
        v = `${v.toLocaleString(undefined,{minimumFractionDigits:2, maximumFractionDigits: 2})}`;

        // Colorear según el tipo de valor
        if (col.includes('Diferencia')) {
          if (v.includes('-')) {
            cellStyle += ' color: #dc3545; font-weight: 600;'; // Rojo para negativo
          } else if (parseFloat(v.replace(/[$,]/g, '')) > 0) {
            cellStyle += ' color: #28a745; font-weight: 600;'; // Verde para positivo
          } else {
            cellStyle += ' color: #6c757d; font-weight: 600;'; // Gris para cero
          }
        } else {
          cellStyle += ' font-weight: 600; color: #495057;';
        }
      } else {
        cellStyle += ' font-weight: 600; color: #495057;';
      }
    }

    if(col === 'Subcategoria' && typeof v === 'string'){
      v = v.toLowerCase().replace(/^\w/, c => c.toUpperCase());
    }

    tbody += `<td style="${cellStyle}">${v}</td>`;
  });
  tbody += '</tr>';
});
tbody += '</tbody>';

tbl.innerHTML = thead + tbody;
tableContainer.appendChild(tbl);
```

```
  c.appendChild(tableContainer);
}

window.addEventListener('DOMContentLoaded', initializeUI);
```