

**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  
PROYECTOS DE NEGOCIO**



**ITESO**  
Universidad Jesuita  
de Guadalajara

**4J05A OPTIMIZACIÓN DE PROGRAMAS DE INVERSIÓN EN INTERMEDIARIOS  
FINANCIEROS**

**Automatización del Análisis de Estados Financieros con Python-Excel**

**PRESENTAN**

**Programas educativos y Estudiantes**

Ingeniería Financiera. de Anda Madrigal, Gustavo

Ingeniería Financiera. Frizard Aguilar, Alejandro

Ingeniería Financiera. Ibarra Delgado, Ricardo

Licenciatura en Finanzas. Morfin Alcalá, Stephany

**Profesor PAP: Sean Nicolás González Vázquez**

**Tlaquepaque, Jalisco, Julio, 2025**

# ÍNDICE

## Contents

|                                                                                  |    |
|----------------------------------------------------------------------------------|----|
| REPORTE PAP .....                                                                | 3  |
| Presentación Institucional de los Proyectos de Aplicación Profesional .....      | 3  |
| Resumen.....                                                                     | 3  |
| 1. Introducción .....                                                            | 5  |
| 2. Desarrollo.....                                                               | 9  |
| • Descripción del proyecto .....                                                 | 10 |
| • Plan de trabajo.....                                                           | 12 |
| • Desarrollo de propuesta de mejora.....                                         | 14 |
| 3. Resultados del trabajo profesional .....                                      | 16 |
| Macro: Importar Balance.....                                                     | 16 |
| Macro: Importar Ejercido .....                                                   | 17 |
| Macro: Limpieza Balance General .....                                            | 17 |
| Macro: Encontrar EvsP.....                                                       | 18 |
| Macro: Join EvsP.....                                                            | 18 |
| Macro: Join BG .....                                                             | 19 |
| Macro: Exportar Dashboard .....                                                  | 19 |
| Script: Procesamiento y División del Balance General.....                        | 20 |
| Script: Estandarización de Cuentas de Activo por Similitud .....                 | 20 |
| Script: Estandarización de Cuentas de Pasivo por Similitud.....                  | 21 |
| Script: Estandarización de Cuentas de Capital por Similitud.....                 | 22 |
| Script: Cálculo de Razones Financieras .....                                     | 23 |
| Script: Identificación y Normalización de Totales del Estado de Resultados ..... | 23 |

|                                                                                                                          |    |
|--------------------------------------------------------------------------------------------------------------------------|----|
| Script: Identificación y Normalización de Subtotales del Estado de Resultados .....                                      | 24 |
| Scripts de Visualización del Estado Ejercido vs Presupuestado .....                                                      | 25 |
| Scripts de Indicadores Financieros Presupuestales .....                                                                  | 26 |
| Cita del código fuente.....                                                                                              | 27 |
| Portada CNIT Estado Financiero Consolidado .....                                                                         | 27 |
| Diapositiva 2.0 .....                                                                                                    | 29 |
| Diapositiva 3.0.....                                                                                                     | 30 |
| 4. Reflexiones de los alumnos sobre sus aprendizajes, las implicaciones éticas y los aportes sociales del proyecto ..... | 32 |
| Aprendizajes profesionales: .....                                                                                        | 32 |
| Aprendizajes sociales:.....                                                                                              | 34 |
| Aprendizajes éticos: .....                                                                                               | 34 |
| Aprendizajes en lo personal: .....                                                                                       | 35 |
| 5. Conclusiones .....                                                                                                    | 36 |
| 6. Bibliografía .....                                                                                                    | 37 |
| Anexos .....                                                                                                             | 38 |

# 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 del desarrollo del proyecto, sus incidencias en el entorno y las reflexiones y aprendizajes profesionales que el estudiante desarrolló durante su labor.*

### Resumen

Este proyecto representa la continuación y mejora del trabajo iniciado en el semestre anterior, enfocado en la automatización del análisis financiero de la Cámara Nacional de la Industria Tequilera (CNIT), mediante el uso de Excel. El objetivo principal del proyecto es optimizar la generación de dashboards que consoliden en tiempo real los datos contables clave como: activos, pasivos, capital, ingresos y egresos, con el fin de apoyar la toma de decisiones estratégicas. La automatización en Python permite reducir errores humanos, aumentar la eficiencia y mejorar la visualización de indicadores clave de desempeño. Para esto se migraron las macros de VBA a Python, conservando únicamente la macro de botones en Excel. Además, se robusteció el código para garantizar su correcto funcionamiento con los datos importados de los estados financieros, mejorando la precisión y eficiencia del procesamiento.

Se optimizó el dashboard financiero, incorporando visualizaciones más completas y análisis avanzados que facilitan la interpretación de indicadores clave como liquidez, rentabilidad y

FO-DGA-CPAP-001

eficiencia operativa. Asimismo, se implementó una nueva funcionalidad que permite exportar los resultados no solo a PDF, sino también a PowerPoint, ofreciendo al usuario la flexibilidad de seleccionar qué gráficas o análisis incluir en la presentación automática.

Este proyecto consolida una herramienta integral que fortalece la toma de decisiones estratégicas en la CNIT, combinando tecnología accesible con un enfoque ético y socialmente responsable.

## 1. Introducción

### 1.1. Objetivos

- **Migrar las macros de VBA a Python** en Excel, manteniendo solo la macro de botones para una interacción amigable con el usuario.
- **Robustecer el código** para garantizar un procesamiento preciso y eficiente de los datos importados de los estados financieros.
- **Mejorar los dashboards financieros**, incorporando visualizaciones más detalladas y análisis avanzados que faciliten la interpretación de KPI's clave.
- **Implementar una función de exportación a PowerPoint**, permitiendo al usuario seleccionar qué gráficas o análisis incluir en la presentación automática.
- **Optimizar la usabilidad**, de la herramienta, asegurando que sea accesible para usuarios no técnicos y adaptable a futuras necesidades de la CNIT.

### 1.2. Justificación

La automatización del análisis financiero en la CNIT representa una necesidad clave en el contexto actual, donde la eficiencia operativa, la transparencia y la agilidad en la toma de decisiones se han vuelto fundamentales para la competitividad institucional. Este proyecto resulta relevante desde una perspectiva disciplinaria, ya que permite aplicar herramientas financieras, tecnológicas y analíticas para resolver una problemática real: la generación manual de reportes contables que puede conllevar errores, retrasos y limitaciones en la interpretación de la información.

Durante el semestre anterior, se desarrolló una solución en Excel con macros en VBA; sin embargo, se identificaron oportunidades de mejora, como la migración a Python para aprovechar su potencia en el manejo de datos y la flexibilidad en la generación de informes. Este proyecto no solo fortalece las capacidades técnicas de la herramienta, sino que también amplía su impacto al incluir funcionalidades avanzadas como la exportación a

PowerPoint, lo que facilita la comunicación de resultados a stakeholders clave.

Desde una perspectiva social, la mejora en la gestión financiera de la CNIT contribuye indirectamente al fortalecimiento de la industria tequilera, un sector emblemático de México. Al fortalecer la capacidad administrativa y estratégica de la CNIT, se contribuye a su misión de impulsar el crecimiento del sector, proteger la denominación de origen y promover mejores condiciones para sus afiliados.

Los beneficios esperados incluyen: una mayor precisión en los reportes financieros, ahorro de tiempo en tareas operativas, visualización clara de indicadores clave, detección oportuna de desviaciones presupuestales y una herramienta duradera que podrá ser replicada o escalada en futuras necesidades de la organización. Asimismo, el proyecto fortalece las competencias profesionales de los estudiantes involucrados, al permitirles enfrentar un reto real con impacto tangible.

### 1.3 Antecedentes

La Cámara Nacional de la Industria Tequilera (CNIT), como organismo representante del sector tequilero en México, realiza múltiples funciones estratégicas, entre ellas la gestión de recursos, la planeación de proyectos y la rendición de cuentas ante sus afiliados. Hasta ahora, muchos de los procesos contables y financieros dentro de la institución se realizaban de manera manual o poco sistematizada, lo cual generaba retrasos en la elaboración de reportes, dificultades para consolidar información y una limitada capacidad para analizar indicadores clave de desempeño de forma ágil y visual.

En el contexto tecnológico actual, donde la automatización y la analítica de datos se han convertido en herramientas fundamentales para la gestión organizacional, se identificó una oportunidad de mejora en los procesos financieros de la CNIT. A partir de esta necesidad, surgió la propuesta de desarrollar una solución en Excel que permitiera estructurar, automatizar y visualizar de manera eficiente la información financiera de la institución.

Este Proyecto de Aplicación Profesional representa la segunda fase de un esfuerzo mayor

por modernizar las herramientas de análisis contable de la CNIT. En el semestre anterior, se realizó un diagnóstico de los estados financieros y del tipo de información disponible; posteriormente, se diseñó una propuesta de dashboards interactivos que integran métricas clave como liquidez, rentabilidad, eficiencia operativa y apalancamiento financiero. Además, se desarrolló una herramienta en Excel que automatizaba el análisis de estados financieros de la CNIT mediante macros en VBA. Si bien esta solución representó un avance significativo, se enfrentaron desafíos como la compatibilidad multiplataforma y limitaciones en el manejo de grandes volúmenes de datos.

Este semestre, el proyecto se enfocó en superar estas limitaciones mediante la migración a Python, un lenguaje más robusto y versátil para el análisis de datos. Además, se mejoró el dashboard para incluir visualizaciones más detalladas y se agregó la funcionalidad de exportación a PowerPoint, respondiendo a las necesidades identificadas en la retroalimentación con los stakeholders. Esta evolución permite sentar las bases para futuras mejoras o integraciones con herramientas más avanzadas, como bases de datos o software contable especializado.

#### 1.4. Contexto

El presente proyecto se desarrolla en el marco de la colaboración con la Cámara Nacional de la Industria Tequilera (CNIT), organismo que representa a las principales empresas productoras de Tequila en México. Fundada en 1959, la CNIT opera como una institución sin fines de lucro que promueve la competitividad, la innovación y la protección de la Denominación de Origen Tequila, uno de los productos más emblemáticos del país tanto a nivel cultural como económico.

La industria tequilera tiene una alta relevancia económica y social, especialmente en el estado de Jalisco y otros estados con denominación de origen. Genera empleos directos e indirectos, impulsa el turismo y representa a México en los mercados internacionales. En este entorno dinámico y altamente regulado, la capacidad de análisis financiero de las organizaciones que integran esta industria se vuelve crucial para garantizar su



sostenibilidad y crecimiento.

Actualmente, la CNIT enfrenta el reto de administrar y presentar su información financiera de forma clara, oportuna y automatizada, como parte de una necesidad creciente por adoptar tecnologías que mejoren la gestión de recursos y la toma de decisiones estratégicas. Este desafío se inserta en un contexto de transformación digital en el que las organizaciones buscan efficientar sus operaciones mediante herramientas tecnológicas accesibles, como Excel, PowerPoint y Python, sin necesidad de implementar sistemas costosos de ERP o software especializado en inteligencia de negocios.

El proyecto surge en respuesta a esta necesidad institucional, en un momento donde la transparencia, la rendición de cuentas y la eficiencia presupuestaria son valores clave para fortalecer la confianza de los afiliados y el posicionamiento institucional de la CNIT dentro del ecosistema económico nacional e internacional.

## 2. Desarrollo

### 2.1. Sustento teórico y metodológico

Desde una perspectiva financiera, el proyecto se basa en la utilización de indicadores clave de desempeño (KPIs), tales como el margen operativo, la liquidez, solvencia, el apalancamiento y el capital de trabajo, que permiten evaluar la salud financiera de una organización (Gitman & Zutter, 2012; Brigham & Ehrhardt, 2013).

Asimismo, se retoman principios contables para la clasificación y análisis de los estados financieros, particularmente el balance general y el estado Presupuestado vs Ejercido, bajo un enfoque estructurado que distingue entre activos, pasivos, capital, ingresos y egresos. Estas categorías son indispensables para el cálculo de razones financieras y para la interpretación eficiente del desempeño institucional.

Desde el punto de vista tecnológico, se emplean fundamentos de automatización y visualización de datos con herramientas de uso común como Microsoft Excel, incluyendo funciones avanzadas, tablas dinámicas, macros y programación con Python. Se migraron las macros de VBA a Python, aprovechando bibliotecas como pandas para el manejo de datos. Estas herramientas permiten estructurar datos de forma eficiente, evitar errores manuales y crear interfaces amigables para el usuario final (Walkenbach, 2015).

Finalmente, la noción de inteligencia de negocios (BI) aporta un enfoque integral al análisis de datos financieros, permitiendo generar dashboards interactivos que transforman información numérica en conocimiento útil para la toma de decisiones estratégicas. Esto implica integrar visualizaciones claras y resúmenes que mejoran la comprensión de la situación financiera de una organización como la CNIT (La Torre & Navarro, 2018).

## 2.2. Planeación y seguimiento del proyecto

- **Descripción del proyecto**

El proyecto inició con la migración completa de las macros de VBA a Python, conservando únicamente la interfaz de botones en VBA para mantener la usabilidad. El proceso incluyó la conversión de todas las funcionalidades clave: importación de datos desde PDF/Excel, limpieza y estandarización de información, cálculo automatizado de indicadores financieros (como liquidez, rentabilidad y análisis vertical), y generación de tablas dinámicas. Para garantizar la precisión, se realizaron pruebas comparativas entre los resultados obtenidos con el código original en VBA y la nueva implementación en Python. Esta propuesta surge como solución a la necesidad de agilizar el análisis contable y presupuestal de la organización, permitiendo un mayor control de sus finanzas, mejor visualización de indicadores clave y reducción de errores humanos en la elaboración de reportes.

Para asegurar la confiabilidad del sistema, se implementaron validaciones que permiten manejar variaciones en los formatos de entrada, como columnas faltantes o cambios en la estructura de los reportes. Se desarrolló un “diccionario” de términos financieros dentro del código para normalizar automáticamente los nombres de cuentas contables, incluso si estos varían entre documentos. El código fue diseñado para identificar dinámicamente partidas clave (activos, pasivos, ingresos) sin depender de posiciones fijas en los archivos. Asimismo, el dashboard fue completamente rediseñado para ofrecer un análisis más completo e intuitivo.

Finalmente, se implementó un botón para exportar resultados a presentaciones de PowerPoint y otro botón para exportarlos a pdf. La herramienta permite seleccionar gráficos específicos del análisis financiero (dashboard), para elaborar la presentación. Esta funcionalidad reduce el tiempo de creación de presentaciones ejecutivas, ahorrando tiempo en la generación de estos reportes.

La pertinencia del proyecto radica en que ofrece una alternativa accesible y personalizable frente a soluciones comerciales de mayor costo, utilizando tecnologías conocidas como

Excel, Power Query, Python y VBA. De esta manera, se logra implementar un sistema robusto que consolida información proveniente de los estados financieros (balance general y estado de resultados), y que permite generar visualizaciones e indicadores clave como liquidez, apalancamiento, eficiencia operativa y rentabilidad del periodo. Cada entrega fue validada con datos simulados que replicaban fielmente los formatos reales de la CNIT. El resultado final es un sistema integral que combina potencia analítica, adaptabilidad y usabilidad para la toma de decisiones financieras.

Bajo esta lógica, el desarrollo se llevó a cabo en distintas etapas:

- **Análisis de requerimientos:** Revisión de la información financiera utilizadas por la CNIT.
- **Migración a Python:** Reemplazo completo de macros VBA por código Python, conservando solo los botones de interfaz.
- **Robustecimiento de código:** Implementación de validaciones para manejar formatos variables.
- **Rediseño de dashboards:** Se agregó el análisis vertical del balance general además de una gráfica de pastel que muestra la ecuación contable de una forma sencilla y digerible.
- **Exportación de información:** Generar reportes automáticos que puedan ser exportados a PDF y a PowerPoint a consideración del usuario.
- **Validación y ajustes:** Pruebas internas y mejoras en la interfaz para asegurar que la herramienta fuera comprensible y útil para los analistas de CNIT.

- **Plan de trabajo**

El plan de trabajo se organizó en torno a entregas quincenales y reuniones de seguimiento cada semana con el profesor asesor. El proyecto se estructuró para desarrollarse en un periodo de 8 semanas, iniciando el 26 de mayo y finalizando el 19 de julio. A lo largo de este periodo, se dividieron las actividades en cuatro entregas principales, permitiendo una evolución progresiva del producto final.

- **Actividades realizadas**

- **Profesionales:** Análisis financiero de los estados contables, interpretación de indicadores y evaluación de necesidades de la CNIT.
- **Técnicas:** Diseño de dashboards en herramientas gráficas (Canva/Illustrator), programación en VBA y Python, y uso de Power Query para importación de datos.
- **Operativas:** Limpieza y transformación de datos, pruebas de funcionalidad, correcciones a partir de retroalimentación.
- **Colaborativas:** Reuniones periódicas con el profesor

- **Recursos necesarios:**

- **Humanos:** Cuatro estudiantes desarrolladores del proyecto, un profesor asesor académico y representantes de la CNIT como stakeholders.
- **Tecnológicos:** Microsoft Excel, Power Query, entorno de macros y Python.
- **Materiales:** Archivos financieros en PDF proporcionados por CNIT, convertidos y trabajados en formato Excel.
- **Temporales:** El calendario de trabajo se distribuyó en tres entregas clave, con reuniones intermedias cada semana para evaluación de avances y toma de decisiones.

- Fechas previstas: de reuniones de los equipos y/o asesorías con el profesor, o bien con otras personas, durante el desarrollo del Proyecto].
  - **Entrega 1:** 16 de mayo, se presentó la definición y alcance de la continuación del proyecto, además de la migración de scripts que se tenían en VBA a Python (incluyendo importación en Excel, limpieza de datos, generación de gráficas y procesamiento). Asimismo, el código se hizo más robusto para evitar problemas con el procesamiento de los datos.
  - **Entrega 2:** 30 de mayo, se hicieron pruebas de carga de archivo y se mejoró el archivo de Excel, eliminando hojas e información innecesaria. Además de presentar un dashboard limpio con únicamente la información necesaria y ya hecho con Python.
  - **Entrega 3:** 14 de julio, se presentó el proyecto final, al dashboard se le agregaron mejoras en el análisis de la información respecto al balance general, agregando gráficas. Asimismo, se agregó la posibilidad de exportación a PowerPoint (ya que previamente solo se podía exportar a pdf), seleccionando, a discreción del usuario, la información a exportar.
  - **Entrega 4:** 19 de julio, se llevó a cabo el evento de presentación final de resultados ante CNIT e ITESO, cuidando la confidencialidad de los datos.

- **Desarrollo de propuesta de mejora**

Dado que este proyecto fue una continuación del desarrollo realizado en el último periodo, a continuación, se muestran las fases desarrolladas tanto el semestre anterior como en este:

*Fase 1: Diagnóstico y diseño conceptual*

Se inició con un análisis detallado del estado presupuestado y del balance general de la CNIT. Esta fase permitió comprender la estructura de los datos disponibles y definir los principales KPIs financieros a representar. Por otro lado, se trabajó en el diseño visual del dashboard, utilizando herramientas como Canva para conceptualizar la interfaz y la disposición de gráficos e indicadores.

*Fase 2: Automatización y prototipo*

Una vez definida la estructura de datos e indicadores, se desarrollaron macros en VBA para importar datos, ejecutar cálculos automáticos y exportar reportes. También se integró Power Query para facilitar la limpieza de datos y su transformación.

Un reto relevante fue la compatibilidad multiplataforma. La primera versión del dashboard fue completamente funcional en Windows, aprovechando funciones específicas para importar PDFs. Al probar en Mac, se identificaron limitaciones importantes, por lo que fue necesario rediseñar el flujo de importación, convirtiendo los archivos PDF a Excel antes de integrarlos a la herramienta.

*Fase 3: Validación*

Se realizaron pruebas funcionales para asegurar el correcto funcionamiento de las macros, la precisión de los indicadores financieros y la integración de los datos importados. Se presentó el avance al stakeholder, quien brindó retroalimentación clave que permitió ajustar detalles visuales y de usabilidad en el dashboard.

*Fase 4: Documentación y Entrega Final*

Se elaboró un manual de uso con instrucciones claras sobre la operación, actualización y solución de errores del dashboard. Posteriormente, se entregó una versión limpia del

archivo y se presentó formalmente el proyecto ante CNIT. Aquí se obtuvo retroalimentación por parte de la CNIT para continuar haciendo mejoras a la herramienta.

#### *Fase 5: Migración Scripts VBA a Python*

Una vez obteniendo retroalimentación, se desarrolló la migración de código VBA a Python para mejorar la eficiencia en el procesamiento de datos y automatización del análisis. Asimismo, se hizo el código más robusto para evitar errores en el procesamiento de estos, cuando el documento origen tuviera detalles que cambiaran.

#### *Fase 6: Mejora del análisis*

La herramienta y los dashboards se mejoraron para que hubiera un análisis integral de la información. Se agregaron gráficas en la parte del balance general para que el análisis fuera más sencillo de entender y fuera más completo. Asimismo, se hizo que la interfaz fuera más sencilla de entender.

#### *Fase 7: Exportación*

Para la exportación de la información, se dejaron dos botones, hechos con macros, donde se puede elegir si exportar el análisis a PDF o a PowerPoint. Además, se buscaba que fuera más amigable, dejando seleccionar al usuario qué información exportar.

#### *Fase 8: Entrega*

Se entregó una versión limpia del archivo y se presentó formalmente el proyecto ante CNIT, concluyendo con éxito la implementación de la herramienta.



### 3. Resultados del trabajo profesional

Como parte del desarrollo del Proyecto de Aplicación Profesional (PAP) en la Cámara Nacional de la Industria Tequilera (CNIT), se creó una herramienta en Excel con funcionalidades automatizadas mediante macros en VBA, que después se migró a **Python en Excel**, orientada al procesamiento, comparación y análisis de información financiera extraída de archivos PDF. Esta solución permitió facilitar el control presupuestal y optimizar los procesos contables, especialmente en los estados financieros de tipo **Balance General** y **Estado de Resultados Ejercido vs Presupuestado**.

A continuación, se detallan lo que se implementó:

#### Macro: Importar Balance

##### Propósito:

Importar desde un archivo Excel (convertido desde PDF) la información del **Balance General**, actualizando el parámetro vinculado en Power Query.

##### Funcionalidad:

- Solicita al usuario seleccionar el archivo fuente del Balance General.
- Actualiza el parámetro de ruta (RutaPDF) utilizado en Power Query.
- Refresca las consultas para cargar automáticamente la información del archivo importado.

## Macro: Importar Ejercido

### Propósito:

Importar el archivo del **Estado de Resultados Ejercido**, también generado desde un PDF convertido a Excel, para su análisis posterior.

### Funcionalidad:

- Permite seleccionar el archivo de resultados ejercidos.
- Actualiza el parámetro RutaPDFRes en Power Query.
- Refresca las consultas para importar los datos del Estado de Resultados Ejercido.

## Macro: Limpieza Balance General

### Propósito:

Estandarizar y extraer las cifras clave del Balance General desde las hojas importadas, para su análisis estructurado.

### Funcionalidad:

- Recorre las hojas Page001 y Page002.
- Busca y extrae automáticamente partidas como Activo, Pasivo y Capital.
- Copia la información en celdas específicas de la hoja de análisis para facilitar su integración al Dashboard.

## Macro: Encontrar EvsP

### Propósito:

Detectar y extraer automáticamente los conceptos relevantes del **Estado de Resultados** tanto **Ejercido como Presupuestado**, directamente desde las hojas importadas.

### Funcionalidad:

- Analiza las hojas Page001 y Page002.
- Localiza conceptos como Ventas, Utilidad Bruta, Gastos, Utilidad Neta, entre otros.
- Extrae las cifras clave y las organiza en la hoja resumen para facilitar la comparación EvsP.

## Macro: Join EvsP

### Propósito:

Unir y organizar en una misma vista las cifras **Ejercidas y Presupuestadas** del Estado de Resultados, permitiendo su comparación directa.

### Funcionalidad:

- Combina ambas versiones del Estado de Resultados en una tabla estructurada.
- Alinea los conceptos similares para que la comparación sea clara y precisa.
- Sirve como base para alimentar visualizaciones y análisis presupuestal.

## Macro: Join BG

### Propósito:

Consolidar las cifras clave del Balance General en un formato uniforme y listo para visualización.

### Funcionalidad:

- Une los datos extraídos del Balance General.
- Los organiza por secciones como Activos, Pasivos y Capital Contable.
- Facilita el análisis mediante indicadores de liquidez, solvencia y estructura financiera.

## Macro: Exportar Dashboard

### Propósito:

Generar una copia consolidada del Dashboard financiero con la información ya procesada, lista para su presentación, archivo o distribución ejecutiva.

### Funcionalidad:

- Crea un nuevo archivo Excel que contiene únicamente las hojas clave del Dashboard, excluyendo hojas auxiliares o de trabajo interno.
- Deja una versión limpia, visual y organizada que puede ser compartida con otras áreas o utilizada para respaldos institucionales.
- Integra dos botones independientes de exportación: Uno para generar un archivo PDF con el dashboard completo o con secciones seleccionadas. Otro para generar una presentación en PowerPoint, donde el usuario puede elegir manualmente qué gráficas o visualizaciones desea incluir, según su criterio y necesidades de comunicación.
- Las exportaciones respetan el formato visual del dashboard: colores, estilos de gráficas y estructura.

### Impacto:

Esta funcionalidad facilita la difusión clara y profesional del análisis financiero, permitiendo que los usuarios obtengan rápidamente reportes listos para consejo directivo, juntas internas o presentaciones externas, sin necesidad de editar manualmente el contenido.

## Script: Procesamiento y División del Balance General

### Propósito:

Automatizar la división de un DataFrame (`df_limpio`) que contiene un estado financiero completo, separándolo en tres DataFrames distintos y estructurados: **Activo**, **Pasivo** y **Capital**, para su análisis individual.

### Funcionalidad:

- **Identifica Marcadores:** Analiza la columna 'Cuenta' para localizar filas clave que contienen las palabras "total" y "suma", usándolas como puntos de división.
- **Extrae Activos:** Crea el DataFrame `df_activo` al seleccionar todas las filas desde el inicio hasta la primera aparición de la palabra "total".
- **Lógica Condicional Adaptativa:** Revisa la fila inmediatamente posterior al total de activos para detectar la palabra "res" (indicativo de una sección de Resultados o Capital).
  - Si encuentra "res": Activa una lógica alternativa que asume que la sección de Capital precede a la de Pasivo. Procede a extraer `df_capital` primero y luego `df_pasivo`.
  - Si no encuentra "res": Aplica la lógica estándar, extrayendo `df_pasivo` (del primer al segundo "total") y luego `df_capital` (después del segundo "total").
- **Limpia Subtotales:** Excluye de los DataFrames finales cualquier fila que contenga la palabra "suma" para evitar la duplicación de datos y mantener solo las cuentas principales.

## Script: Estandarización de Cuentas de Activo por Similitud

### Propósito:

Homologar las cuentas del `df_activo` que pueden tener nombres variables o con errores tipográficos, asociándolas a una lista de cuentas de referencia estándar. Este proceso, conocido como "fuzzy matching" o mapeo por similitud, garantiza que los datos puedan ser agrupados y analizados de forma consistente.

### Funcionalidad:

- **Define Cuentas de Referencia:** Establece una lista maestra con los nombres correctos y deseados para las cuentas de activo (ej. "Caja", "Bancos", "Deudores Diversos").

- **Función de Normalización:** Utiliza una función (`normalize_text`) para limpiar tanto las cuentas del archivo original como las de referencia. Este paso elimina diferencias por mayúsculas, puntuación o espacios extra.
- **Comparación por Similitud:** Itera sobre cada cuenta del DataFrame original y la compara contra toda la lista de referencia, calculando un porcentaje de similitud textual para cada par.
- **Identifica la Mejor Coincidencia:** Para cada cuenta original, determina cuál de las cuentas de referencia es la más parecida y guarda su puntuación.
- **Aplica Umbral de Confianza:** Solo considera válidas aquellas coincidencias que superan un umbral de similitud predefinido (80%), descartando mapeos poco probables.
- **Genera DataFrame Estandarizado:** Crea un nuevo DataFrame final que contiene únicamente las cuentas que fueron exitosamente mapeadas, mostrando el nombre original, el nombre de referencia asignado, el porcentaje de coincidencia y sus valores numéricos correspondientes.

#### Script: Estandarización de Cuentas de Pasivo por Similitud

##### Propósito:

Homologar las cuentas del `df_pasivo` que pueden tener nombres con variaciones o errores. Este proceso asocia cada cuenta original a una cuenta estándar de una lista de referencia, utilizando una comparación por similitud de texto. El objetivo es asegurar la consistencia de los datos para su correcta agrupación y análisis.

##### Funcionalidad:

- **Define Cuentas de Referencia:** Establece una lista maestra con los nombres oficiales para las cuentas de pasivo (ej. "Proveedores", "Impuestos Retenidos", "Acreedores diversos a corto plazo").
- **Función de Normalización:** Emplea una función (`normalize_text`) para limpiar y estandarizar los nombres de las cuentas. Este paso es crucial, ya que elimina diferencias por mayúsculas, puntuación o espaciado irregular, permitiendo una comparación más precisa.
- **Comparación por Similitud:** Recorre cada cuenta del DataFrame original y la compara contra toda la lista de referencia, calculando un porcentaje de similitud textual (`SequenceMatcher`) para encontrar el mejor candidato.
- **Identifica la Mejor Coincidencia:** Para cada cuenta original, el script identifica la

cuenta de referencia más parecida y su puntuación de similitud.

- **Aplica Umbral de Confianza:** Descarta mapeos poco probables al considerar como válidas únicamente aquellas coincidencias que superan un umbral de similitud del **80%**.
- **Genera DataFrame Estandarizado:** Crea y devuelve un nuevo DataFrame (`df_pasivo`) que contiene exclusivamente las cuentas que fueron mapeadas con éxito. El resultado muestra el nombre original, el nombre de referencia asignado, el porcentaje de coincidencia y sus valores financieros.

### Script: Estandarización de Cuentas de Capital por Similitud

**Propósito:** Homologar y estandarizar las cuentas contables dentro del `df_capital`. Este proceso utiliza un algoritmo de comparación de texto para mapear nombres de cuentas con variaciones o errores tipográficos a una lista de cuentas de referencia, asegurando la integridad y consistencia de los datos para su posterior análisis financiero.

**Funcionalidad:**

- **Define Cuentas de Referencia:** Establece una lista maestra con los nombres oficiales y deseados para las cuentas de capital (ej. "Resultado Ejercicios Anteriores", "Remanente o Deficit del Ejercicio").
- **Función de Normalización:** Utiliza una función (`normalize_text`) para limpiar tanto las cuentas del archivo de origen como las de la lista de referencia. Este paso elimina diferencias por mayúsculas, acentos, puntuación o espaciado para permitir una comparación efectiva.
- **Comparación por Similitud:** Itera sobre cada cuenta del DataFrame original, la normaliza y la compara contra toda la lista de referencia normalizada para calcular un porcentaje de similitud textual (`SequenceMatcher`).
- **Identifica la Mejor Coincidencia:** Para cada cuenta original, el script determina cuál de las cuentas de referencia es textualmente más parecida y registra su puntuación de similitud.
- **Aplica Umbral de Confianza:** Filtra los resultados, considerando como válidas únicamente las coincidencias que alcanzan o superan un umbral de similitud del **80%**. Esto previene mapeos incorrectos o poco fiables.
- **Genera DataFrame Estandarizado:** Crea y devuelve un nuevo DataFrame (`df_capital`) que contiene solo las cuentas mapeadas con éxito. El resultado final incluye el nombre de la cuenta original, el nombre de la cuenta de referencia asignada, el porcentaje de coincidencia y sus valores financieros asociados.

## Script: Cálculo de Razones Financieras

**Propósito:** Calcular un conjunto de indicadores y razones financieras clave a partir de los DataFrames ya limpios y estandarizados (`df_activo`, `df_pasivo`, `df_capital`). El objetivo es convertir los datos contables en métricas de rendimiento para evaluar la liquidez, endeudamiento y estructura financiera de la entidad.

**Funcionalidad:**

- **Calcula la Razón Circulante:** Mide la capacidad de la empresa para cubrir sus deudas a corto plazo. Se calcula dividiendo el **activo total** entre el **pasivo total**.
- **Calcula la Razón de Efectivo (Prueba Ácida Estricta):** Determina la capacidad de pago inmediata de la empresa sin depender de la venta de inventarios. Suma las cuentas más líquidas (**Caja, Bancos e Inversiones**) y las divide entre el **pasivo total**.
- **Calcula la Capacidad de Endeudamiento:** Indica el porcentaje de los activos que está financiado a través de deuda. Se calcula dividiendo el **pasivo total** entre el **activo total**.
- **Calcula el Apalancamiento Financiero:** Mide la relación entre el capital ajeno (deuda) y el capital propio. Se calcula dividiendo el **pasivo total** entre el **capital total**.
- **Calcula la Autonomía Financiera:** Representa la independencia financiera de la empresa frente a sus acreedores. Se calcula dividiendo el **capital total** entre el **pasivo total**.

## Script: Identificación y Normalización de Totales del Estado de Resultados

**Propósito:**

Detectar, limpiar y estandarizar las partidas que representan totales o subtotales relevantes dentro del Estado Ejercido vs Presupuestado, aun cuando estas aparezcan con nombres inconsistentes o genéricos como “Total”.

**Funcionalidad:**

- Lee el rango original de texto importado desde Excel, que contiene las descripciones contables (por ejemplo, “Total”, “TOTAL GASTOS”, etc.).
- Aplica una función de limpieza que normaliza los textos: elimina caracteres especiales, convierte a minúsculas y filtra solo palabras significativas.
- Crea una columna auxiliar (`Texto_Limpio`) para trabajar con descripciones uniformes.
- Detecta automáticamente las filas que contienen la palabra “total” y excluye las que



contienen “subtotal” para evitar duplicados o errores de interpretación.

- Agrupa las filas por bloques delimitados por estos totales, identificando a qué sección pertenecen.
- Si una fila de total tiene la descripción ambigua “total” (sin especificar a qué corresponde), se le asigna automáticamente un nombre más completo concatenando el primer concepto del grupo al que pertenece (por ejemplo, “Total Gastos Operativos”).
- Genera una tabla final (df\_totales) que contiene únicamente las partidas de totales ya normalizadas, listas para visualización y análisis.

#### Impacto:

Este script mejora la consistencia y claridad de los totales contables, incluso cuando provienen de formatos PDF convertidos a Excel que pueden tener textos mal estructurados o incompletos. Es especialmente útil para asegurar que los dashboards muestren etiquetas claras y comparables entre archivos.

#### Script: Identificación y Normalización de Subtotales del Estado de Resultados

##### Propósito:

Detectar, estandarizar y limpiar las partidas que representan **subtotales** dentro del Estado Ejercido vs Presupuestado, especialmente aquellas que aparecen de forma genérica (como “Subtotal”) o con textos inconsistentes, para asegurar su correcta interpretación y visualización.

##### Funcionalidad:

Lee el rango de datos original desde Excel, que incluye las descripciones de las cuentas contables.

- Aplica una función de limpieza textual que convierte todo a minúsculas, elimina caracteres especiales y conserva únicamente palabras significativas.
- Crea una columna auxiliar Texto\_Limpio con las descripciones normalizadas para facilitar el procesamiento.
- Detecta las filas que contienen la palabra “subtotal” mediante expresiones regulares.
- Crea grupos de filas contiguas según la aparición de subtotales, permitiendo identificar a qué sección pertenece cada subtotal.
- Extrae únicamente las filas que representan subtotales.
- En los casos donde el texto es simplemente “subtotal”, el script sustituye ese término por uno más completo, agregando automáticamente la primera descripción

significativa del grupo al que pertenece (por ejemplo, “subtotal ingresos ordinarios”).

- Estandariza los textos de subtotales para que comiencen siempre con “subtotal”, sin repeticiones o espacios innecesarios.
- Devuelve un DataFrame final limpio (df\_subtotales) con todos los subtotales corregidos y preparados para visualización.

#### Impacto:

Este script garantiza que los subtotales del Estado de Resultados se presenten de forma clara y consistente, independientemente de cómo hayan sido escritos en el archivo fuente. Esto permite alimentar correctamente los dashboards comparativos y asegura que cada subtotal sea interpretado de forma unívoca en los reportes de control presupuestal.

#### [Scripts de Visualización del Estado Ejercido vs Presupuestado](#)

##### Propósito:

Generar visualizaciones claras y comparativas que permitan analizar rápidamente la relación entre lo presupuestado y lo ejercido en distintas partidas del Estado de Resultados. Estas gráficas están diseñadas para facilitar la interpretación visual de desviaciones, cumplimiento y distribución de montos.

##### Visualizaciones generadas:

- **Gráfico de barras apiladas (Totales):**  
Muestra la distribución porcentual entre lo presupuestado y lo ejercido para los totales contables principales. Se utiliza un formato horizontal 100% apilado que permite visualizar cuánto representa cada monto respecto al total combinado. Las etiquetas incluyen los montos en pesos y están diseñadas para facilitar la lectura ejecutiva.
- **Gráfico de dispersión con regresión logarítmica (Subtotales):**  
Representa cada subtotal como un punto en función del monto presupuestado (eje X, escala logarítmica) y el monto ejercido (eje Y), permitiendo visualizar visualmente qué partidas están sobreejercidas o subejercidas. Se incluye una línea de referencia (regresión) y etiquetas con flechas que identifican cada partida, ayudando a detectar rápidamente desviaciones presupuestales significativas.
- **Gráfico de barras apiladas (Subtotales):**  
Similar al gráfico de totales, pero aplicado a los subtotales contables. Permite observar cómo se distribuye el cumplimiento presupuestal por cada grupo funcional o categoría contable, ofreciendo una lectura sintética de las áreas con mejor o peor ejecución relativa.

#### Impacto:

Estas visualizaciones son clave dentro del dashboard financiero, ya que transforman los datos en información visual que puede ser interpretada fácilmente por usuarios no técnicos.

Facilitan la toma de decisiones y la detección oportuna de desviaciones, cumpliendo con el objetivo de hacer el análisis presupuestal más intuitivo y accionable para la CNIT.

### Scripts de Indicadores Financieros Presupuestales

#### Propósito:

Mostrar de forma clara, visual y ejecutiva los **indicadores clave de desempeño presupuestal**, directamente calculados a partir de los totales del Estado de Resultados. Cada indicador se presenta en una tarjeta visual (caja) con sombra, color institucional y texto central, optimizando su lectura en reportes ejecutivos y dashboards.

#### Indicadores generados:

- **Ingresos Contra Presupuesto (% de cumplimiento):**  
Compara los ingresos ejercidos contra el presupuesto asignado. Calcula el porcentaje de cumplimiento y lo muestra como una cifra central en formato “XX.XX %”. Útil para evaluar el nivel de captación presupuestal.
- **Ingresos Ejercidos (valor absoluto):**  
Muestra el monto total ejercido en ingresos, en formato monetario. Este valor sirve como referencia rápida del tamaño operativo actual de la institución.
- **Egresos Ejercidos (valor absoluto):**  
Detecta automáticamente la fila que contiene el total de egresos en el dataset, incluso si la nomenclatura cambia entre archivos. Muestra el valor ejercido como cifra principal.
- **Egresos Contra Presupuesto (% de cumplimiento):**  
Calcula el porcentaje de cumplimiento del presupuesto en egresos. Este indicador es fundamental para detectar posibles sobreejercicios o subejercicios.
- **Proporción de Gasto Operativo:**  
Calcula qué porcentaje de los egresos totales corresponde al gasto operativo. Ayuda a entender la estructura del gasto institucional y el peso relativo de los gastos fijos o recurrentes.
- **Balance Presupuestal (Ingresos – Egresos):**  
Muestra el resultado neto entre ingresos y egresos ejercidos. Puede interpretarse como superávit o déficit presupuestal, y se presenta en valor monetario con formato legible para ejecutivos.

#### Impacto:

Estas tarjetas financieras permiten comunicar de forma ágil y estandarizada los principales KPIs presupuestales, facilitando la evaluación rápida de la ejecución financiera. Su diseño visual está optimizado para dashboards, presentaciones a consejo y documentos internos de

control presupuestal.

#### Cita del código fuente

El código fuente completo de cada macro o script descrito se encuentra disponible en el **Apéndice A** de este documento, donde puede consultarse con fines de revisión técnica o reproducción del proyecto (véase Apéndice A).

#### Resultados Dashboard exportado

##### Portada CNIT Estado Financiero Consolidado



## Diapositiva 1.0



# Resumen Balance General

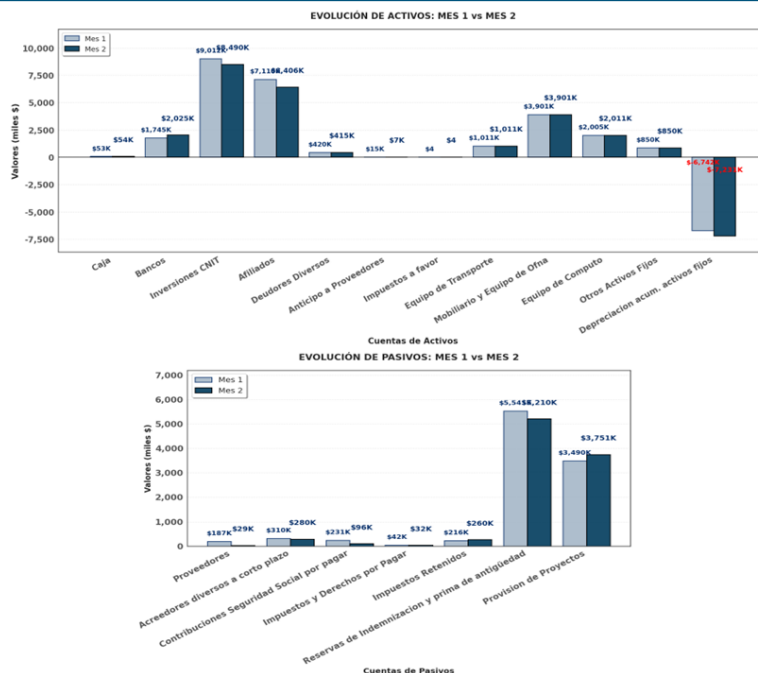
**Razón Circulante**  
**1.86**

**Razón de Efectivo**  
**1.09**

**Autonomía Financiera**  
**1.26**

**Capacidad de Endeudamiento**  
**53.84 %**

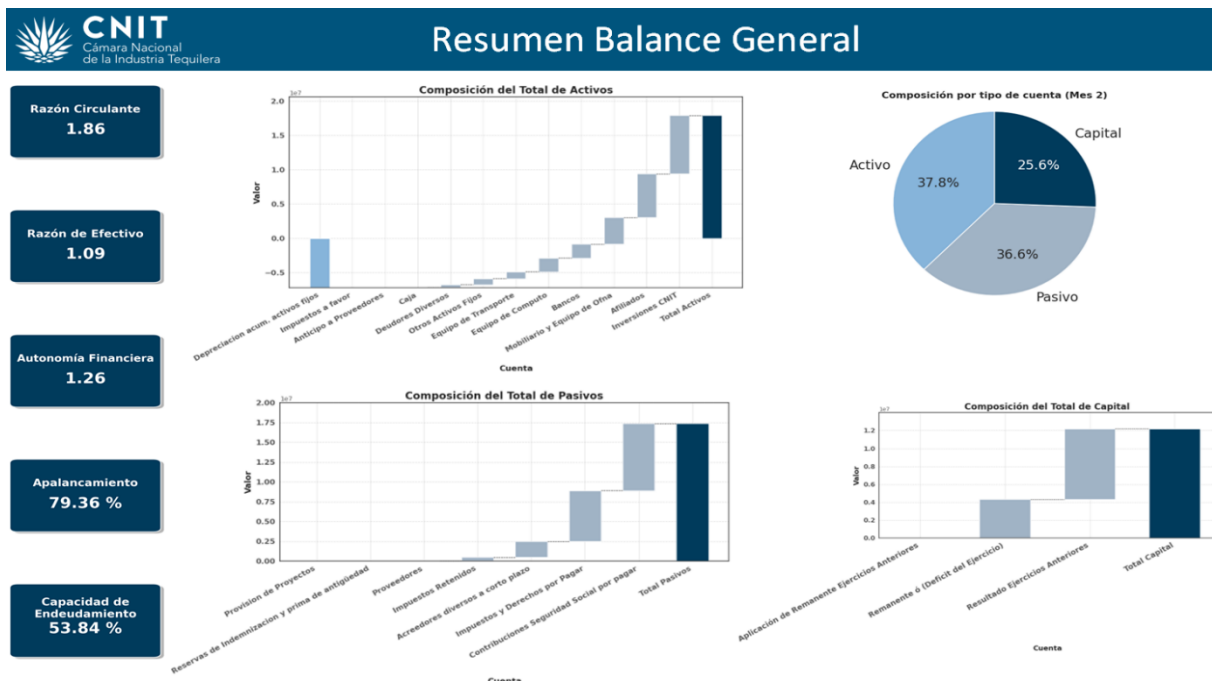
**Apalancamiento**  
**79.36 %**



*Los datos mostrados son simulados por razones de confidencialidad conforme al NDA firmado con CNIT.*

La diapositiva sobre el "Resumen Balance General" presenta una visualización clara de la situación financiera de la empresa. En el lado derecho, se muestran dos gráficos de barras comparativos. El superior detalla la "Evolución de Activos" entre el Mes 1 y el Mes 2, desglosando cuentas como caja, bancos e inversiones. El gráfico inferior sigue el mismo formato para la "Evolución de Pasivos", mostrando rubros como proveedores e impuestos. En el costado izquierdo, se destacan cinco indicadores financieros clave, cada uno en un recuadro individual para un rápido diagnóstico: Razón Circulante, Razón de Efectivo, Autonomía Financiera, Capacidad de Endeudamiento y Apalancamiento. Este formato permite una interpretación eficiente, ya que combina un análisis detallado de las cuentas del balance con los indicadores de desempeño más relevantes, facilitando la toma de decisiones estratégicas al ofrecer un panorama completo de la liquidez y la estructura de endeudamiento de la compañía.

## Diapositiva 2.0



**Los datos mostrados son simulados por razones de confidencialidad conforme al NDA firmado con CNIT.**

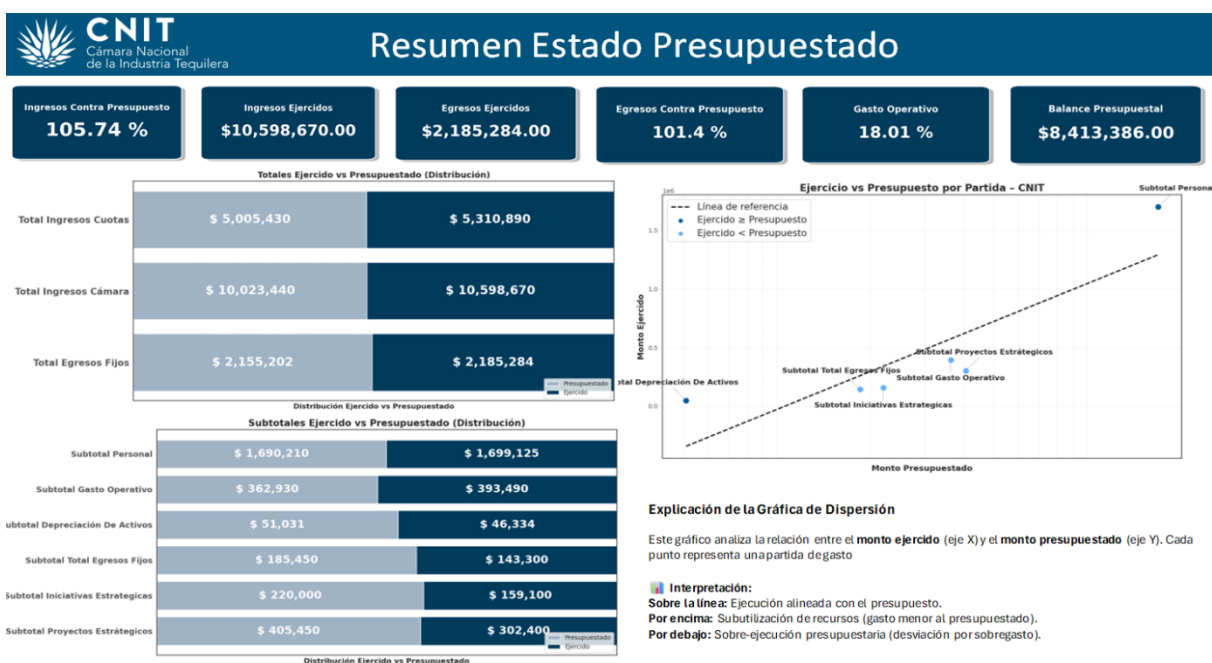
La diapositiva 2.0, titulada "Resumen Balance General", ofrece un análisis detallado de la composición financiera de la empresa. En el lado derecho, se presentan dos visualizaciones clave: un gráfico circular que desglosa la estructura del capital, mostrando la proporción de Activo, Pasivo y Capital, y un gráfico de cascada que detalla la composición del capital total. En la parte central, se utilizan gráficos de cascada para ilustrar la "Composición del Total de Activos" y la "Composición del Total de Pasivos". Estos gráficos muestran cómo se acumulan las distintas cuentas, desde las de menor valor hasta las de mayor valor, para conformar los totales.

Al igual que en la diapositiva anterior, en el costado izquierdo se mantienen los cinco indicadores financieros clave en recuadros individuales para una referencia rápida: Razón Circulante, Razón de Efectivo, Autonomía Financiera, Apalancamiento y Capacidad de Endeudamiento.

Este formato facilita una comprensión profunda de la estructura del balance general,

permitiendo identificar rápidamente qué cuentas tienen mayor peso en los activos, pasivos y capital, a la vez que se mantienen visibles los indicadores de rendimiento más importantes.

### Diapositiva 3.0



*Los datos mostrados son simulados por razones de confidencialidad conforme al NDA firmado con CNIT.*

La diapositiva 3.0, "Resumen Estado Presupuestado", está diseñada para ofrecer un análisis completo de la ejecución presupuestaria. En la parte superior, una franja con seis recuadros destaca los indicadores más importantes, como el cumplimiento de ingresos y egresos contra el presupuesto, los montos totales ejercidos y el balance presupuestal, brindando un resumen ejecutivo inmediato.

El cuerpo principal de la izquierda presenta una serie de gráficos de barras horizontales que comparan los montos ejercidos contra los presupuestados para distintas categorías y subtotales de ingresos y egresos. Este formato permite una revisión detallada de las principales áreas, identificando fácilmente dónde se produjeron las variaciones.

A la derecha, se incluye un gráfico de dispersión que analiza la relación entre el monto ejercido

y el presupuestado para cada partida individual. Este elemento visual se complementa con una explicación detallada sobre cómo interpretar el gráfico, indicando si una partida está alineada, por encima o por debajo del presupuesto.

Esta estructura visual guía al usuario desde un panorama general (indicadores superiores) hacia un análisis detallado (barras a la izquierda) y finalmente a una evaluación por partida específica (gráfico de dispersión), facilitando una comprensión integral del desempeño presupuestario.



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

##### **Aprendizajes profesionales:**

Durante el desarrollo del Proyecto de Aplicación Profesional (PAP), se fortalecieron competencias clave tanto genéricas como disciplinares, propias de los programas de Ingeniería Financiera y Finanzas. El proyecto implicó la aplicación práctica de conocimientos técnicos en análisis financiero, automatización de procesos y visualización de información, a través del diseño de una herramienta en Excel con apoyo de Power Query, Visual Basic for Applications (VBA) y Python.

La experiencia permitió consolidar habilidades de pensamiento crítico, trabajo colaborativo, gestión de proyectos y comunicación efectiva. Se integraron saberes de contabilidad, finanzas corporativas, programación y diseño de indicadores financieros, lo que permitió abordar de forma integral una necesidad concreta dentro de una organización del sector industrial.

Además, el equipo de trabajo desarrolló una comprensión más amplia del entorno económico e institucional en el que opera la Cámara Nacional de la Industria Tequilera (CNIT), lo que contribuyó a una mayor conciencia sobre el papel de los organismos intermedios en el fortalecimiento de sectores estratégicos del país. El proyecto permitió observar cómo el conocimiento académico puede trasladarse eficazmente a entornos reales, generando soluciones concretas y de utilidad inmediata.

### **Importancia de aprender y actualizarse Python en Excel:**

El aprendizaje y la actualización en Python dentro del entorno de Excel representan una evolución estratégica en el perfil de cualquier profesionalista que trabaje con datos, finanzas o procesos operativos. Tradicionalmente, Excel ha sido una herramienta indispensable en las organizaciones, pero su integración con Python abre nuevas posibilidades de automatización, análisis avanzado y personalización que superan las capacidades de las macros en VBA.

En el contexto de este proyecto, Python permitió automatizar procesos complejos, hacer el código más robusto frente a cambios de formato, y generar dashboards inteligentes con visualizaciones dinámicas. Además, al tratarse de un lenguaje de programación ampliamente adoptado en la industria, aprender Python en Excel brinda portabilidad de habilidades hacia otros entornos como bases de datos, software de inteligencia de negocios o incluso desarrollo web.

La capacidad de usar Python directamente en Excel (a través de `=PY()` y otras funciones integradas en la versión moderna) permite combinar lo mejor de ambos mundos: la familiaridad de Excel con el poder de procesamiento de Python. Esto significa mayor eficiencia, menor probabilidad de errores, y sobre todo, mayor escalabilidad para soluciones que antes eran exclusivamente manuales.

Finalmente, dominar esta integración prepara a los estudiantes y profesionistas para responder a las demandas de un mercado laboral que valora cada vez más las habilidades analíticas, de automatización y de ciencia de datos, sin abandonar las herramientas que ya son estándar en muchas industrias.

**Aprendizajes sociales:**

El PAP brindó la oportunidad de comprender cómo la mejora de procesos técnicos internos dentro de una organización puede derivar en beneficios sociales más amplios. La herramienta desarrollada puede optimizar el análisis financiero de la CNIT, lo que afecta positivamente en la toma de decisiones y en la rendición de cuentas de una institución que representa a múltiples actores de la industria tequilera.

Se identificó que los saberes aplicados —automatización, indicadores financieros, visualización de datos— son transferibles a otros contextos institucionales y pueden ser replicados en diferentes sectores. Esto permite ampliar el alcance del impacto social del proyecto, al promover la eficiencia operativa, la transparencia y la sostenibilidad en organizaciones similares.

A través del trabajo con la CNIT, se reconoció que la generación de valor social no está limitada a proyectos con enfoque comunitario directo, sino que también puede darse mediante herramientas que fortalezcan el funcionamiento de instituciones clave para el desarrollo económico regional y nacional.

**Aprendizajes éticos:**

Durante el desarrollo del proyecto, se tomaron decisiones que respondieron no solo a criterios técnicos, sino también éticos. Una de las principales fue optar por una solución basada en herramientas de bajo costo y fácil acceso, priorizando la sostenibilidad del sistema propuesto, su autonomía tecnológica y la posibilidad de que el equipo interno de la CNIT pudiera operarlo sin depender de proveedores externos.

Se consideraron principios como la honestidad, la responsabilidad profesional, el respeto por los tiempos y procesos de la organización, y la comunicación clara con los distintos actores involucrados. La experiencia permitió reflexionar sobre el papel que deben asumir las y los profesionistas en la generación de soluciones que respondan a necesidades reales y que se construyan en colaboración con quienes las van a utilizar.

Este enfoque ético fue parte integral del diseño y ejecución del proyecto, y se considera como una base fundamental para futuros desempeños profesionales de los integrantes del equipo.

### **Aprendizajes en lo personal:**

El PAP representó un espacio de crecimiento y aprendizaje integral. Permitió a los integrantes del equipo reconocer sus capacidades, asumir roles complementarios, identificar áreas de oportunidad y fortalecer habilidades interpersonales en contextos reales. La experiencia ofreció un acercamiento directo al funcionamiento de una institución empresarial, lo que permitió dimensionar los retos técnicos y humanos del trabajo profesional.

El proyecto favoreció el desarrollo de la autonomía, la iniciativa y el compromiso, al tener que planear, ejecutar y evaluar un proyecto de forma estructurada y con objetivos concretos. Asimismo, brindó una experiencia significativa de trabajo colaborativo, donde se pusieron en práctica valores como la corresponsabilidad, la escucha activa y el respeto por la diversidad de ideas.

Este proceso contribuyó a consolidar la identidad profesional de quienes participaron, al evidenciar que las herramientas adquiridas durante la formación académica pueden tener un impacto positivo más allá del aula, y al abrir nuevas perspectivas sobre los ámbitos en los que pueden desarrollarse como profesionistas en el futuro.

## 5. Conclusiones

Este proyecto representa la segunda etapa de una iniciativa más amplia para modernizar y automatizar el análisis financiero de la Cámara Nacional de la Industria Tequilera (CNIT). En esta fase, se logró cumplir de manera efectiva con los objetivos planteados, desarrollando una herramienta automatizada en Excel que permite importar, transformar, analizar y exportar los estados financieros de forma estructurada, eficiente y visualmente clara. El sistema resultante demostró ser funcional, accesible y adaptable a las necesidades específicas del área financiera de la CNIT. Asimismo, se entregó un manual de uso que garantiza su continuidad operativa más allá del cierre del proyecto.

Un logro central de esta segunda etapa fue la migración del código desde macros en VBA hacia Python integrado en Excel, lo cual permitió robustecer los procesos, reducir el margen de error, y mejorar la capacidad de adaptación ante cambios en los formatos de los estados financieros. Esta actualización tecnológica no solo mejora el rendimiento y la escalabilidad de la herramienta, sino que también representa una competencia clave para los profesionistas involucrados, alineada con las tendencias actuales de automatización en el análisis financiero.

No obstante, se identificó una limitante importante: la funcionalidad de Python en Excel solo está disponible para usuarios de Windows, mientras que la versión para MacOS aún se encuentra en fase beta. Esta situación implica que la herramienta aún no es completamente multiplataforma, lo cual debe considerarse en futuras mejoras o despliegues institucionales más amplios.

El Proyecto de Aplicación Profesional fue un ejemplo exitoso de colaboración entre universidad y empresa, permitiendo a los estudiantes aplicar conocimientos técnicos, éticos y profesionales en un contexto real. Más allá de resolver una necesidad puntual, esta segunda fase sienta las bases para que la CNIT continúe mejorando e innovando en la gestión de su información financiera mediante soluciones tecnológicas sostenibles y escalables.

## 6. Bibliografía

CNIT | Cámara Nacional de la Industria Tequilera | México. (2025, 5 febrero). CNIT.  
<https://www.cnit.org.mx/>

Brigham, E. F., & Ehrhardt, M. C. (2013). *Fundamentals of Financial Management* (13th ed.). Cengage Learning.

Gitman, L. J., & Zutter, C. J. (2012). *Principios de administración financiera* (12.ª ed.). Pearson Educación.

Walkenbach, J. (2015). *Excel 2016 Bible: The Comprehensive Tutorial Resource*. Wiley.

La Torre, C., & Navarro, M. (2018). *Tableros de control y visualización de datos: Power BI, Excel y otras herramientas*. Ediciones Díaz de Santos.

## 7. Anexos

### Código Visual Basic:

#### *Macro: ExportarPDF*

```
Public Sub ExportDashBoardtoPDF(selItems As Variant)
    Dim pptApp As Object
    Dim pptPres As Object
    Dim pptSlide As Object
    Dim rng As Range
    Dim shtNames As Variant, ranges As Variant
    Dim i As Long, parts As Variant
    Dim wsName As String, picName As String
    Dim ws As Worksheet, sh As Shape
    Dim pdfFile As String
    Dim tplRng As Range
    Dim tplImg As Object
    Dim alturaTpl As Single
    Dim grafShp As Object
    Dim tituloShape As Object

    ' 1) Inicializar PowerPoint
    On Error Resume Next
    Set pptApp = GetObject(Class:="PowerPoint.Application")
    If pptApp Is Nothing Then Set pptApp =
CreateObject(Class:="PowerPoint.Application")
    On Error GoTo 0
    pptApp.Visible = True
    Set pptPres = pptApp.Presentations.Add
    Application.ScreenUpdating = False

    ' 2) Copiar rangos a slides...
    shtNames = Array("Portada", "BG Graficas", "BG Graficas2", "EvsP Graficas")
    ranges = Array("A1:O30", "E1:S30", "E1:S30", "E1:S30")
    For i = 0 To UBound(shtNames)
        With ThisWorkbook.Worksheets(shtNames(i))
            If .Visible <> xlSheetVisible Then .Visible = xlSheetVisible
            Set rng = .Range(ranges(i))
        End With
        rng.CopyPicture xlScreen, xlPicture
        Set pptSlide = pptPres.Slides.Add(pptPres.Slides.Count + 1, 12)
        pptSlide.Shapes.Paste
        With pptSlide.Shapes(pptSlide.Shapes.Count)
            .LockAspectRatio = msoFalse
            .Left = 0: .Top = 0
            .Width = pptPres.PageSetup.SlideWidth
            .Height = pptPres.PageSetup.SlideHeight
        End With
        If shtNames(i) = "Portada" Then ThisWorkbook.Worksheets("Portada").Visible
= xlSheetHidden
    Next i

    ' 3) Pegar las formas seleccionadas...
    For i = LBound(selItems) To UBound(selItems)
        FO-DGA-CPAP-001
```

```

parts = Split(selItems(i), "|")
wsName = parts(0)
picName = parts(1)
Set ws = ThisWorkbook.Worksheets(wsName)
Set sh = ws.Shapes(picName)

' 1) Nueva diapositiva en blanco
Set pptSlide = pptPres.Slides.Add(pptPres.Slides.Count + 1, 12)

' 2) Pegar la plantilla estirada a TODO el slide
With ThisWorkbook.Worksheets("Template")
    .Visible = xlSheetVisible          ' asegúrate de que está visible
    .Range("A1:O30").CopyPicture xlPrinter, xlPicture
    .Visible = xlSheetHidden          ' si la quieres oculta de nuevo
End With
Set tplImg = pptSlide.Shapes.Paste() (1)
With tplImg
    .LockAspectRatio = msoFalse
    .Left = 0
    .Top = 0
    .Width = pptPres.PageSetup.SlideWidth
    .Height = pptPres.PageSetup.SlideHeight
    .ZOrder msoSendToBack          ' lo manda detrás
End With

' 3) Pegar título (TextBox "Titulo") encima de la plantilla
' — 3) Añadir title box dinámico y centrarlo —
Dim titleBox As Object
Set titleBox = pptSlide.Shapes.AddTextbox( _
    Orientation:=msoTextOrientationHorizontal, _
    Left:=0, _
    Top:=10, _
    Width:=pptPres.PageSetup.SlideWidth, _
    Height:=30)

With titleBox.TextFrame.TextRange
    .Text = picName          ' el nombre de la gráfica
    .Font.Name = "Calibri"   ' opcional: fuente
    .Font.Size = 30          ' opcional: tamaño
    .Font.Color.RGB = RGB(255, 255, 255)
    .ParagraphFormat.Alignment = 2 ' centrar en horizontal
End With

titleBox.ZOrder msoBringToFront

' 4) Pegar el gráfico encima de la plantilla
sh.CopyPicture xlPrinter, xlPicture
Set grafShp = pptSlide.Shapes.Paste() (1)
With grafShp
    .LockAspectRatio = msoTrue

    ' — guardar tamaño original —
    Dim origW As Single, origH As Single
    origW = .Width

```



```

origH = .Height

' — calcular área disponible —
Dim maxW As Single, maxH As Single
maxW = pptPres.PageSetup.SlideWidth * 0.8
maxH = pptPres.PageSetup.SlideHeight _
      - (titleBox.Top + titleBox.Height + 10) _
      - 20      ' 20 pts de colchón inferior

' — escalar proporcionalmente para que no sobrepase ni ancho ni alto —
Dim scaleFactor As Double
scaleFactor = WorksheetFunction.Min(maxW / origW, maxH / origH)
.Width = origW * scaleFactor
.Height = origH * scaleFactor

' — centrar debajo del título —
.Left = (pptPres.PageSetup.SlideWidth - .Width) / 2
.Top = titleBox.Top + titleBox.Height + 10

.ZOrder msoBringToFront
End With

Next i

' 4) Exportar a PDF con SaveAs
pdfFile = ThisWorkbook.Path & "\DashBoard_" & Format(Date, "yyyy-mm-dd") &
".pdf"
pptPres.SaveAs pdfFile, 32      ' 32 = ppSaveAsPDF

' 5) Cerrar PPT sin guardar el .pptx y restaurar pantalla
pptPres.Saved = True
pptPres.Close
pptApp.Quit
Set pptPres = Nothing
Set pptApp = Nothing
Application.ScreenUpdating = True

MsgBox "PDF generado en:" & vbCrLf & pdfFile, vbInformation
End Sub

```

### *Macro: ExportarPPP1*

```

Public Sub ExportDashBoardAndPics(selItems As Variant)
    Dim pptApp As Object
    Dim pptPres As Object
    Dim pptSlide As Object
    Dim rng As Range
    Dim shtNames As Variant, ranges As Variant
    Dim i As Long, parts As Variant
    Dim wsName As String, picName As String
    Dim ws As Worksheet, sh As Shape
    Dim tplSht As Worksheet, tplImg As Object
    Dim titleBox As Object, grafShp As Object
    Dim origW As Single, origH As Single
    Dim maxW As Single, maxH As Single

```

```

Dim scaleFac As Double
Dim fileName As String

' --- 1) Inicializar PowerPoint ---
On Error Resume Next
Set pptApp = GetObject(Class:="PowerPoint.Application")
If pptApp Is Nothing Then Set pptApp =
CreateObject(Class:="PowerPoint.Application")
On Error GoTo 0
pptApp.Visible = True
Set pptPres = pptApp.Presentations.Add
Application.ScreenUpdating = False

' Referencia a la hoja Template
Set tplSht = ThisWorkbook.Worksheets("Template")

' --- 2) Exportar Portada y hojas fijas SIN plantilla ---
shtNames = Array("Portada", "BG Graficas", "BG Graficas2", "EvsP Graficas")
ranges = Array("A1:O30", "E1:S30", "E1:S30", "E1:S30")
For i = 0 To UBound(shtNames)
    With ThisWorkbook.Worksheets(shtNames(i))
        If .Visible <> xlSheetVisible Then .Visible = xlSheetVisible
        Set rng = .Range(ranges(i))
    End With

    rng.CopyPicture Appearance:=xlScreen, Format:=xlPicture
    Set pptSlide = pptPres.Slides.Add(pptPres.Slides.Count + 1, 12)
    pptSlide.Shapes.Paste
    With pptSlide.Shapes(pptSlide.Shapes.Count)
        .LockAspectRatio = msoFalse
        .Left = 0
        .Top = 0
        .Width = pptPres.PageSetup.SlideWidth
        .Height = pptPres.PageSetup.SlideHeight
    End With

    ' Si era la Portada, volver a ocultarla
    If shtNames(i) = "Portada" Then ThisWorkbook.Worksheets("Portada").Visible
= xlSheetHidden
Next i

' --- 3) Exportar las imágenes seleccionadas CON plantilla ---
For i = LBound(selItems) To UBound(selItems)
    parts = Split(selItems(i), "|")      ' ["Hoja","NombreShape"]
    wsName = parts(0)
    picName = parts(1)
    Set ws = ThisWorkbook.Worksheets(wsName)
    Set sh = ws.Shapes(picName)

    ' 3.a) Nueva diapositiva en blanco
    Set pptSlide = pptPres.Slides.Add(pptPres.Slides.Count + 1, 12)

```

```

' 3.b) Pegar la plantilla (TODO el slide)
With tplSht
    .Visible = xlSheetVisible
    .Range("A1:O30").CopyPicture xlPrinter, xlPicture
    .Visible = xlSheetHidden
End With
Set tplImg = pptSlide.Shapes.Paste()(1)
With tplImg
    .LockAspectRatio = msoFalse
    .Left = 0
    .Top = 0
    .Width = pptPres.PageSetup.SlideWidth
    .Height = pptPres.PageSetup.SlideHeight
    .ZOrder msoSendToBack
End With

' 3.c) Añadir título dinámico centrado
Set titleBox = pptSlide.Shapes.AddTextbox( _
    Orientation:=msoTextOrientationHorizontal, _
    Left:=0, _
    Top:=10, _
    Width:=pptPres.PageSetup.SlideWidth, _
    Height:=30)
With titleBox.TextFrame.TextRange
    .Text = picName
    .Font.Name = "Calibri"
    .Font.Size = 30
    .Font.Color.RGB = RGB(255, 255, 255)
    .ParagraphFormat.Alignment = 2 ' centro
End With
titleBox.ZOrder msoBringToFront

' 3.d) Copiar y pegar la gráfica como imagen de alta calidad
sh.CopyPicture xlPrinter, xlPicture
Set grafShp = pptSlide.Shapes.Paste()(1)
With grafShp
    .LockAspectRatio = msoTrue

    ' Tamaño original
    origW = .Width: origH = .Height
    ' Espacio disponible
    maxW = pptPres.PageSetup.SlideWidth * 0.8
    maxH = pptPres.PageSetup.SlideHeight _
        - (titleBox.Top + titleBox.Height + 10) _
        - 20
    ' Factor de escala
    scaleFac = WorksheetFunction.Min(maxW / origW, maxH / origH)
    .Width = origW * scaleFac
    .Height = origH * scaleFac

    ' Centrar horizontalmente y colocar debajo del título
    .Left = (pptPres.PageSetup.SlideWidth - .Width) / 2

```

```

        .Top = titleBox.Top + titleBox.Height + 10

        .ZOrder msoBringToFront
    End With
Next i

' --- 4) Guardar la presentación ---
fileName = "DashBoard_" & Format(Date, "yyyy-mm-dd") & ".pptx"
pptPres.SaveAs ThisWorkbook.Path & "\" & fileName

Application.ScreenUpdating = True
MsgBox "Presentación guardada en:" & vbCrLf & ThisWorkbook.Path & "\" &
fileName, vbInformation
End Sub

```

### *Macro: Importar\_BG*

```

Option Explicit

Public rutaBGSeleccionada As String

' Procedimiento principal que el usuario ejecutará
Sub SeleccionarBGyActualizarPowerQuery()
    Dim rutaArchivo As String
    rutaArchivo = SeleccionarArchivoBG()

    ' Verificar si se seleccionó un archivo
    If rutaArchivo = "" Then
        MsgBox "No se seleccionó ningún archivo. Operación cancelada.",
vbInformation
        Exit Sub
    End If

    rutaBGSeleccionada = rutaArchivo

    ActualizarParametroPowerQuery "RutaBG", rutaArchivo

    MsgBox "Ruta actualizada en Power Query:" & vbCrLf & rutaArchivo, vbInformation

    ActualizarTodasLasConsultas
End Sub

' Funcion corregida compatible con Excel Mac
Function SeleccionarArchivoBG() As String
    Dim rutaArchivo As Variant

    #If Mac Then
        rutaArchivo = Application.GetOpenFilename()
        If rutaArchivo = False Then rutaArchivo = ""
    #Else
        With Application.FileDialog(msoFileDialogFilePicker)

```

FO-DGA-CPAP-001

```

        .Title = "Seleccionar archivo Excel"
        .Filters.Clear
        .Filters.Add "Archivos Excel", "*.xls;*.xlsx;*.xlsm"
        .AllowMultiSelect = False

        If .Show = -1 Then
            rutaArchivo = .SelectedItem(1)
        Else
            rutaArchivo = ""
        End If
    End With
#End If

    SeleccionarArchivoBG = rutaArchivo
End Function

Sub ActualizarParametroPowerQuery(nombreParametro As String, valorParametro As String)
    On Error Resume Next

    valorParametro = Replace(valorParametro, "\", "/")
    ThisWorkbook.Queries(nombreParametro).Formula = """" & valorParametro & """"

    If Err.Number <> 0 Then
        Err.Clear
        CrearParametroPowerQuery nombreParametro, valorParametro
    End If

    On Error GoTo 0
End Sub

Sub CrearParametroPowerQuery(nombreParametro As String, valorParametro As String)
    MsgBox "El parámetro '" & nombreParametro & "' no existe." & vbCrLf & _
        "Por favor, créalo manualmente desde Power Query.", vbInformation
End Sub

Sub ActualizarTodasLasConsultas()
    On Error Resume Next

    Application.StatusBar = "Actualizando consultas de Power Query..."

    ThisWorkbook.RefreshAll

    Application.Wait Now + TimeValue("00:00:02")

    Application.StatusBar = False

    If Err.Number <> 0 Then
        MsgBox "Error al actualizar consultas: " & Err.Description, vbExclamation
        Err.Clear
    End If

    On Error GoTo 0

```

```

End Sub

Sub CrearBotonSeleccionBG()
    Dim btn As Button
    Dim hoja As Worksheet

    Set hoja = ActiveSheet

    On Error Resume Next
    hoja.Buttons("btnSeleccionarBG").Delete
    On Error GoTo 0

    Set btn = hoja.Buttons.Add(100, 20, 150, 30)

    With btn
        .Name = "btnSeleccionarBG"
        .Caption = "Seleccionar Excel Balance General"
        .OnAction = "SeleccionarBGyActualizarPowerQuery"
    End With

    MsgBox "Botón creado correctamente.", vbInformation
End Sub

```

```

Function ObtenerRutaBG() As String
    ObtenerRutaBG = rutaBGSeleccionada
End Function

```

### *Macro: Importar\_EvsP*

```

Option Explicit

Public rutaResultados As String

' Procedimiento principal que el usuario ejecuta
Sub SeleccionaryActualizarPQRes()
    Dim rutaArchivoRes As String
    rutaArchivoRes = SeleccionarArchivoRes()

    ' Verificar si se seleccionó un archivo
    If rutaArchivoRes = "" Then
        MsgBox "No se seleccionó ningún archivo. Operación cancelada.",
vbInformation
        Exit Sub
    End If

    rutaResultados = rutaArchivoRes

    ActualizarParametroPQRes "RutaRes", rutaArchivoRes

    MsgBox "Ruta actualizada en Power Query:" & vbCrLf & rutaArchivoRes,
vbInformation

    ActualizarTodasLasConsultasRes

```

```

End Sub
FO-DGA-CPAP-001

```

```

' Función corregida compatible con Excel Mac
Function SeleccionarArchivoRes() As String
    Dim rutaArchivo As Variant

    #If Mac Then
        rutaArchivo = Application.GetOpenFilename()
        If rutaArchivo = False Then rutaArchivo = ""
    #Else
        With Application.FileDialog(msoFileDialogFilePicker)
            .Title = "Seleccionar archivo Excel"
            .Filters.Clear
            .Filters.Add "Archivos Excel", "*.xls;*.xlsx;*.xlsm"
            .AllowMultiSelect = False

            If .Show = -1 Then
                rutaArchivo = .SelectedItems(1)
            Else
                rutaArchivo = ""
            End If
        End With
    #End If

    SeleccionarArchivoRes = rutaArchivo
End Function

Sub ActualizarParametroPQRes(nombreParametroRes As String, valorParametroRes As String)
    On Error Resume Next

    valorParametroRes = Replace(valorParametroRes, "\", "/")
    ThisWorkbook.Queries(nombreParametroRes).Formula = """" & valorParametroRes & """"

    If Err.Number <> 0 Then
        Err.Clear
        CrearParametroPQRes nombreParametroRes, valorParametroRes
    End If

    On Error GoTo 0
End Sub

Sub CrearParametroPQRes(nombreParametroRes As String, valorParametroRes As String)
    MsgBox "El parámetro '" & nombreParametroRes & "' no existe." & vbCrLf & _
        "Por favor, créalo manualmente desde Power Query.", vbInformation
End Sub

Sub ActualizarTodasLasConsultasRes()
    On Error Resume Next

    Application.StatusBar = "Actualizando consultas de Power Query..."

    ThisWorkbook.RefreshAll

    Application.Wait Now + TimeValue("00:00:02")

```

```

Application.StatusBar = False

If Err.Number <> 0 Then
    MsgBox "Error al actualizar consultas: " & Err.Description, vbExclamation
    Err.Clear
End If

On Error GoTo 0
End Sub

Sub CrearBotonSeleccionRes()
    Dim btnRes As Button
    Dim hojaRes As Worksheet

    Set hojaRes = ActiveSheet

    On Error Resume Next
    hojaRes.Buttons("btnSeleccionarRes").Delete
    On Error GoTo 0

    Set btnRes = hojaRes.Buttons.Add(100, 20, 150, 30)

    With btnRes
        .Name = "btnSeleccionarRes"
        .Caption = "Seleccionar Excel Ejercido vs. Presupuestado"
        .OnAction = "SeleccinaryActualizarPQRes"
    End With

    MsgBox "Botón creado correctamente.", vbInformation
End Sub

Function ObtenerRutaRes() As String
    ObtenerRutaRes = rutaResultados
End Function

```

### *Macro: Selector*

```

'- variable que indica a qué flujo vamos
Public g_ExportMode As String

'- llama al formulario pero marcando modo PPT
Public Sub ShowPPTSelector()
    g_ExportMode = "PPT"
    frmPicSelector.Show
End Sub

'- llama al formulario pero marcando modo PDF
Public Sub ShowPDFSelector()
    g_ExportMode = "PDF"
    frmPicSelector.Show
End Sub

```



## Script: Procesamiento y División del Balance General

```
cuentas_lower = df_limpio['Cuenta'].str.lower() total_indices =  
cuentas_lower[cuentas_lower.str.contains('total', na=False)].index.tolist() suma_indices_a_excluir =  
cuentas_lower[cuentas_lower.str.contains('suma', na=False)].index.tolist() idx_total_1 =  
total_indices[0] if len(total_indices) > 0 else -1  
  
if idx_total_1 != -1: df_activo_raw = df_limpio.iloc[0:idx_total_1] df_activo =  
df_activo_raw[~df_activo_raw.index.isin(suma_indices_a_excluir)] else: df_activo =  
pd.DataFrame(columns=df_limpio.columns)  
  
df_pasivo = pd.DataFrame(columns=df_limpio.columns) df_capital =  
pd.DataFrame(columns=df_limpio.columns)  
  
if idx_total_1 != -1 and (idx_total_1 + 1) < len(df_limpio) and  
'res' in cuentas_lower.iloc[idx_total_1 + 1]:  
  
start_capital = idx_total_1 + 1  
search_space_capital = cuentas_lower.iloc[start_capital:]  
stop_mask_capital = search_space_capital.str.contains('suma', na=False)  
next_suma_indices = stop_mask_capital[stop_mask_capital].index.tolist()  
  
idx_suma_stop = -1  
if len(next_suma_indices) > 0:  
    idx_suma_stop = next_suma_indices[0]  
    df_capital_raw = df_limpio.iloc[start_capital : idx_suma_stop]  
    df_capital = df_capital_raw[~df_capital_raw.index.isin(suma_indices_a_excluir)]  
  
if idx_suma_stop != -1:  
    start_pasivo = -1  
    for i in range(idx_suma_stop, len(df_limpio)):  
        if 'suma' not in cuentas_lower.iloc[i]:  
            start_pasivo = i  
            break  
  
if start_pasivo != -1:  
    search_space_pasivo = cuentas_lower.iloc[start_pasivo:]  
    stop_mask_pasivo = search_space_pasivo.str.contains('total', na=False)  
    next_total_indices = stop_mask_pasivo[stop_mask_pasivo].index.tolist()  
  
if len(next_total_indices) > 0:  
    idx_total_stop = next_total_indices[0]  
    df_pasivo_raw = df_limpio.iloc[start_pasivo : idx_total_stop]  
    df_pasivo = df_pasivo_raw[~df_pasivo_raw.index.isin(suma_indices_a_excluir)]  
else:  
    df_pasivo_raw = df_limpio.iloc[start_pasivo:]  
    df_pasivo = df_pasivo_raw[~df_pasivo_raw.index.isin(suma_indices_a_excluir)]
```

```

else: idx_total_2 = total_indices[1] if len(total_indices) > 1 else -1

if idx_total_1 != -1 and idx_total_2 != -1:
    df_pasivo_raw = df_limpio.iloc[idx_total_1 + 1 : idx_total_2]
    df_pasivo = df_pasivo_raw[~df_pasivo_raw.index.isin(suma_indices_a_excluir)]

if idx_total_2 != -1:
    search_space_capital = cuentas_lower.iloc[idx_total_2 + 1:]
    stop_mask_capital = search_space_capital.str.contains('suma', na=False)
    next_suma_indices = stop_mask_capital[stop_mask_capital].index.tolist()

    if len(next_suma_indices) >= 2:
        idx_final_stop = next_suma_indices[1]
        df_capital_raw = df_limpio.iloc[idx_total_2 + 1 : idx_final_stop]
        df_capital = df_capital_raw[~df_capital_raw.index.isin(suma_indices_a_excluir)]
    else:
        df_capital_raw = df_limpio.iloc[idx_total_2 + 1 :]
        df_capital = df_capital_raw[~df_capital_raw.index.isin(suma_indices_a_excluir)]

```

df\_activo

### Script: Estandarización de Cuentas de Activo por Similitud

```

df_activo = xl("E3")

def normalize_text(text): if pd.isna(text): return "" text = str(text).lower() text = re.sub(r'[\w\s]', '',
text) text = re.sub(r'\s+', ' ', text).strip() return text

reference_accounts_activo = [ "Caja", "Bancos", "Inversiones CNIT", "Afiliados", "Deudores Diversos",
"Anticipo a Proveedores", "Impuestos a favor", "Equipo de Transporte", "Mobiliario y Equipo de
Ofna", "Equipo de Computo", "Otros Activos Fijos", "Depreciacion acum. activos fijos" ]

normalized_reference_accounts = [normalize_text(acc) for acc in reference_accounts_activo]

SIMILARITY_THRESHOLD = 0.80

result_list = []

for index, row in df_activo.iterrows(): account_name_original = row['Cuenta']
normalized_account_name_df = normalize_text(account_name_original)

best_match_name = None
best_match_score = 0.0

for i, ref_account_norm in enumerate(normalized_reference_accounts):

```

```

matcher = SequenceMatcher(None, normalized_account_name_df, ref_account_norm)
current_score = matcher.ratio()

if current_score > best_match_score:
    best_match_score = current_score
    best_match_name = reference_accounts_activo[i]

if best_match_score >= SIMILARITY_THRESHOLD:
    result_list.append({
        'Cuenta Original DF': account_name_original,
        'Cuenta de Referencia': best_match_name,
        'Porcentaje Coincidencia': round(best_match_score * 100, 2),
        'Mes 1': row['Valor 1'],
        'Mes 2': row['Valor 2']
    })

df_activo = pd.DataFrame(result_list)

df_activo

```

### Script: Estandarización de Cuentas de Pasivo por Similitud

```

df_pasivo = xl("G3")

def normalize_text(text): if pd.isna(text): return "" text = str(text).lower() text = re.sub(r'^\w\s', "", text) text = re.sub(r'\s+', ' ', text).strip() return text

reference_accounts_pasivo = [ "Proveedores", "Acreedores diversos a corto plazo", "Contribuciones Seguridad Social por pagar", "Impuestos y Derechos por Pagar", "Impuestos Retenidos", "Reservas de Indemnizacion y prima de antigüedad", "Provision de Proyectos" ]

normalized_reference_accounts_pasivo = [normalize_text(acc) for acc in reference_accounts_pasivo]

SIMILARITY_THRESHOLD = 0.80

result_list_pasivo = []

for index, row in df_pasivo.iterrows(): account_name_original = row['Cuenta']
normalized_account_name_df = normalize_text(account_name_original)

best_match_name = None
best_match_score = 0.0

```

```

for i, ref_account_norm in enumerate(normalized_reference_accounts_pasivo):
    matcher = SequenceMatcher(None, normalized_account_name_df, ref_account_norm)
    current_score = matcher.ratio()

    if current_score > best_match_score:
        best_match_score = current_score
        best_match_name = reference_accounts_pasivo[i]

if best_match_score >= SIMILARITY_THRESHOLD:
    result_list_pasivo.append({
        'Cuenta Original DF': account_name_original,
        'Cuenta de Referencia': best_match_name,
        'Porcentaje Coincidencia': round(best_match_score * 100, 2),
        'Mes 1': row['Valor 1'],
        'Mes 2': row['Valor 2']
    })

df_pasivo = pd.DataFrame(result_list_pasivo)

df_pasivo

```

#### Script: Estandarización de Cuentas de Capital por Similitud

```

df_capital = xl("I3")

def normalize_text(text): if pd.isna(text): return "" text = str(text).lower() text = re.sub(r'^\w\s', '',
text) text = re.sub(r'\s+', ' ', text).strip() return text

reference_accounts_capital = [ "Resultado Ejercicios Anteriores", "Aplicación de Remanente
Ejercicios Anteriores", "Remanente o Deficit del Ejercicio" ]

normalized_reference_accounts_capital = [normalize_text(acc) for acc in
reference_accounts_capital]

SIMILARITY_THRESHOLD = 0.80

result_list_capital = []

for index, row in df_capital.iterrows(): account_name_original = row['Cuenta']
normalized_account_name_df = normalize_text(account_name_original)

best_match_name = None
best_match_score = 0.0

for i, ref_account_norm in enumerate(normalized_reference_accounts_capital):
    matcher = SequenceMatcher(None, normalized_account_name_df, ref_account_norm)

```

```

current_score = matcher.ratio()

if current_score > best_match_score:
    best_match_score = current_score
    best_match_name = reference_accounts_capital[i]

if best_match_score >= SIMILARITY_THRESHOLD:
    result_list_capital.append({
        'Cuenta Original DF': account_name_original,
        'Cuenta de Referencia': best_match_name,
        'Porcentaje Coincidencia': round(best_match_score * 100, 2),
        'Mes 1': row['Valor 1'],
        'Mes 2': row['Valor 2']
    })

```

```
df_capital = pd.DataFrame(result_list_capital)
```

```
df_capital
```

### Script: Cálculo de Razones Financieras

--- Razón Circulante ---

```
total_activos = df_activo['Mes 2'].sum() total_pasivos = df_pasivo['Mes 2'].sum() r_circulante =
total_activos / total_pasivos r_circulante
```

--- Razón de Efectivo (Prueba Súper Ácida) ---

```
caja = df_activo[df_activo['Cuenta de Referencia'] == 'Caja']['Mes 2'].sum() bancos =
df_activo[df_activo['Cuenta de Referencia'] == 'Bancos']['Mes 2'].sum() inversiones =
df_activo[df_activo['Cuenta de Referencia'] == 'Inversiones CNIT']['Mes 2'].sum() efectivo_y_eq =
caja + bancos + inversiones r_efectivo = efectivo_y_eq / total_pasivos r_efectivo
```

--- Razón de Endeudamiento ---

```
cap_endeudamiento = (total_pasivos / total_activos) * 100 cap_endeudamiento
```

--- Razón de Apalancamiento Financiero ---

```
total_capital = df_capital['Mes 2'].sum() apalancamiento = (total_pasivos / total_capital) * 100
```

--- Razón de Autonomía Financiera ---

```
aut_financiera = (total_capital / total_pasivos) aut_financiera
```

## Script: Identificación y Normalización de Totales del Estado de Resultados

```
import pandas as pd
import re

# Leer rango desde Excel
df = xl("B2")

# Función para limpiar texto
def limpiar_texto(texto):
    texto_str = str(texto).lower() if pd.notna(texto) else ""
    palabras_limpias = " ".join(re.findall(r'\b[a-záéíóúüñ]+\b', texto_str))
    return palabras_limpias

# Crear columna Texto_Limpio
df['Texto_Limpio'] = df.iloc[:,0].apply(limpiar_texto).str.lower()

# Detectar filas que contienen 'total' y no contienen 'subtotal'
df['es_total'] = df['Texto_Limpio'].str.contains(r'\btotal\b', na=False) &
~df['Texto_Limpio'].str.contains('subtotal', na=False)

# Crear grupos
df['Grupo'] = df['es_total'].cumsum()

# Lista para guardar filas finales
filas_total_finales = []

# Procesar cada grupo
for _, grupo_df in df.groupby('Grupo'):
    grupo_df = grupo_df.copy()

    # Obtener el primer valor significativo del grupo
    primera_desc = grupo_df.iloc[0, 0] # columna original

    for idx, row in grupo_df.iterrows():
        if row['es_total']:
            # Si el texto limpio es exactamente "total", se concatena la descripción
            if row['Texto_Limpio'].strip() == 'total':
                df.at[idx, 'Texto_Limpio'] = f"total {limpiar_texto(primera_desc)}"

    # Agregar solo las filas que son totales
    filas_total_finales.append(grupo_df[grupo_df['es_total']])

# Concatenar todos los subtotales corregidos
df_totales = pd.concat(filas_total_finales, ignore_index=True)
```

```
df_totales
```

### Script: Identificación y Normalización de Subtotales del Estado de Resultados

```
import pandas as pd
import re
import numpy as np

# Leer el rango desde Excel (ajusta según tu archivo)
df = xl("B2")

# Función para limpiar texto y dejar solo palabras (minúsculas)
def limpiar_texto(texto):
    texto_str = str(texto).lower() if pd.notna(texto) else ""
    palabras_limpias = " ".join(re.findall(r'\b[a-záéíóúñ]+\b', texto_str))
    return palabras_limpias

# Crear columna Texto_Limpio
df['Texto_Limpio'] = df.iloc[:, 0].apply(limpiar_texto).str.lower()

# Identificar subtotales
df['es_subtotal'] = df['Texto_Limpio'].str.contains(r'\bsubtotal\b', na=False)

# Crear grupos
df['Grupo'] = df['es_subtotal'].cumsum() - df['es_subtotal'].astype(int)
# Eliminar columna auxiliar
df.drop(columns=['es_subtotal'], inplace=True)
# Obtener primeras descripciones por grupo
primeras_descripciones = df.groupby('Grupo')['Texto_Limpio'].first().to_dict()
# Identificar subtotales únicamente
df_subtotales = df[df['Texto_Limpio'].str.contains('subtotal', na=False)].copy()
# Reemplazar los subtotales incompletos
df_subtotales['Texto_Limpio'] = df_subtotales.apply(
    lambda row: f'subtotal {primeras_descripciones[row['Grupo']]}' if row['Texto_Limpio'].strip() ==
'subtotal' else row['Texto_Limpio'],
    axis=1
)
# Limpiar el texto: asegurar que empiece por "subtotal", quitar duplicados o espacios extra
df_subtotales['Texto_Limpio'] = df_subtotales['Texto_Limpio'].str.replace(r'\bsubtotal\b\s*', '',
regex=True)
df_subtotales['Texto_Limpio'] = 'subtotal ' + df_subtotales['Texto_Limpio']
df_subtotales['Texto_Limpio'] = df_subtotales['Texto_Limpio'].str.replace(r'\s+', ' ',
regex=True).str.strip()

# Resultado final: solo subtotales con descripciones limpias
df_subtotales.reset_index(drop=True, inplace=True)
```

df\_subtotales