

Universidad de las Américas
Facultad de Ingeniería y Ciencias Aplicadas

Integración de Sistemas
Progreso 1 – Evaluación Práctica

Estudiante: Iván Tulcán

Contents

Indicaciones	3
Solución	4
Proceso a seguir para transferir la información mensual del sistema contable a la base de datos de respaldo.	4
Diagrama del Proceso	5
Implementación de la solución.....	5
Descripción del Sistema y Componentes	5
Proceso de Respaldo de la Información	7
Validación y Control de Errores.....	8
Evidencia de Implementación.....	8
Repositorio de la solución:.....	14

Indicaciones

Considerando que el Sistema Contable no tiene Soporte alguno y el mismo contiene la información financiera de toda la historia de la Cooperativa, es un riesgo que en algún momento el Sistema deje de funcionar y no se lo pueda recuperar. Por lo que es necesario ir generando un respaldo de la información crítica del mismo que actualmente se la puede obtener de reportes mensuales del Sistema, a una Base de Datos alojada en los servidores de la Institución. Para el caso podemos empezar respaldando la información del reporte de flujo de efectivo que tiene los siguientes campos, , , , . (Ej. 8-11- 2023, 11, 2023, Ingreso, 4,500)

- Describa el proceso que se seguirá para transferir la información mensual del Sistema contable a la Base de datos de Respaldo
- Realice el diagrama de flujo que se seguirá para transferir la información mensual del Sistema contable a la Base de datos de Respaldo
- Implemente la solución
- En su implementación considera la validación y control de posibles errores, data inconsistente, data duplicada.

Solución

Proceso a seguir para transferir la información mensual del sistema contable a la base de datos de respaldo.

Considerando que el sistema legacy “Sistema Contable” permite tanto el registro de nuevos ingresos/egresos en su base de datos, como también la generación de reportes, se plantea el siguiente proceso:

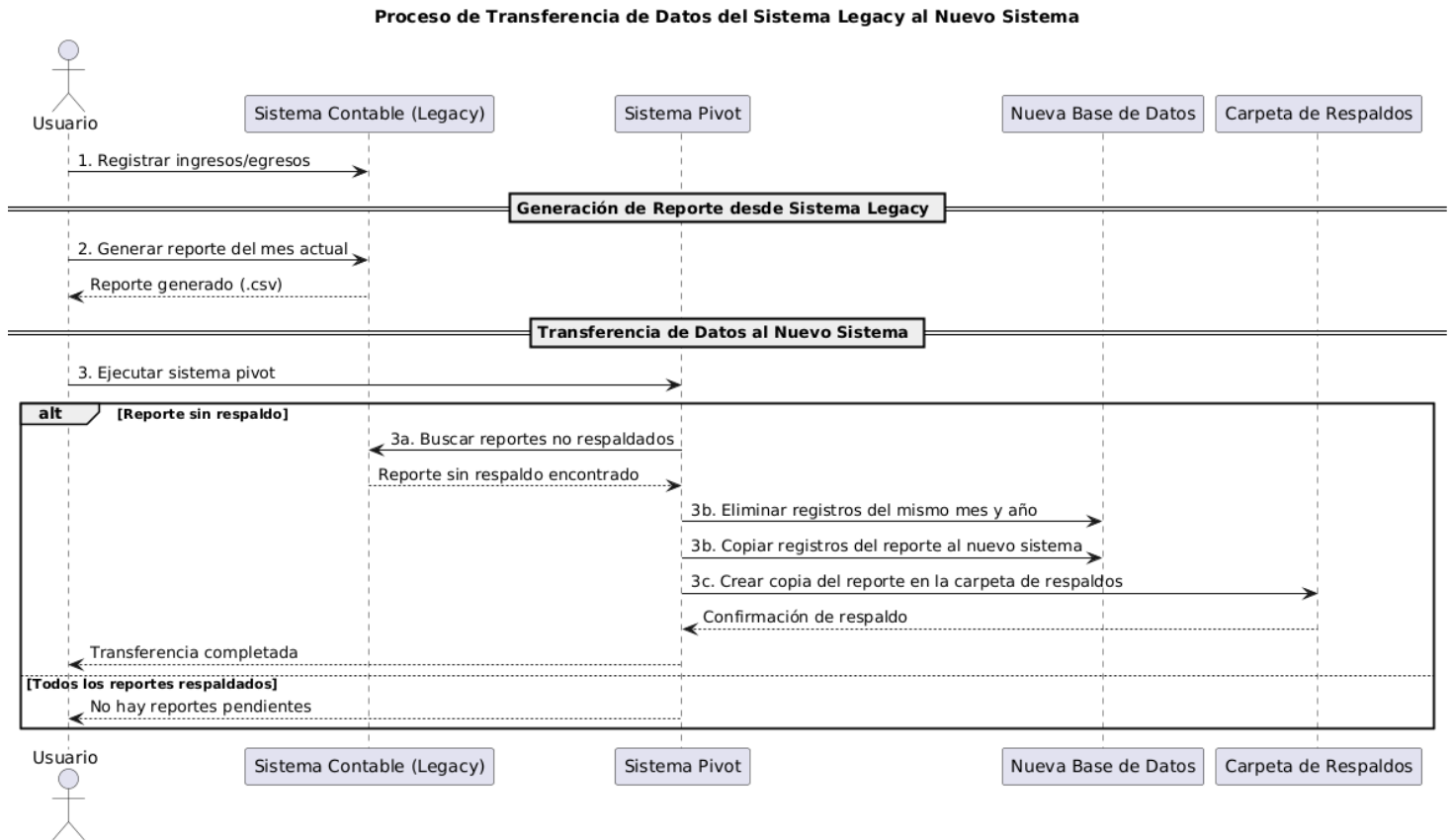
1. Usuario registra ingresos/egresos a través del sistema Legacy.
2. El usuario responsable de la transferencia de datos desde el sistema legacy hacia el nuevo, genera un reporte del mes actual desde el sistema Legacy.
3. El mismo usuario del paso anterior ejecuta un sistema pivot que se encargará de:

- a. Buscar los reportes generados por el sistema legacy que no se hayan respaldado en la nueva base de datos.
- b. Al encontrar un reporte que no ha sido copiado en la nueva base de datos, procederá a copiar cada registro de dicho reporte en la nueva base de datos.

NOTA: El sistema deberá eliminar todos los registros existentes en la nueva base de datos que coincidan con el rango de fechas del reporte para evitar duplicados.

- c. Al terminar de copiar todos los registros del reporte en la nueva base de datos, creará una copia del reporte (.csv) en una carpeta auxiliar. Esto permitirá llevar un registro de los reportes que ya han sido respaldados en la nueva base de datos.

Diagrama del Proceso.



Implementación de la solución.

Este sistema fue diseñado para responder a las necesidades de respaldo y manejo de información contable de la Cooperativa "Cajita S.A." y sigue un flujo de trabajo que cubre desde la entrada de registros de ingresos/egresos hasta la generación de reportes históricos. La implementación es modular, y cada script cumple una función específica en el flujo del proceso.

Descripción del Sistema y Componentes

Este sistema consta de tres scripts principales que colaboran en conjunto:

1. **SistemaContable.py:**

- **Funcionalidad:** Este script es la interfaz principal para registrar manualmente transacciones (ingresos o egresos) o generar registros

aleatorios de manera rápida, con el fin de simular datos. Estos registros se almacenan en una base de datos SQLite llamada BDDLegacy.sqlite.

- **Rol en el Sistema:** Su función es ser la fuente de datos inicial, permitiendo tanto el registro manual como la generación masiva de registros de prueba. También incluye la capacidad de exportar estos registros en formato .csv (un archivo de reporte mensual) y guardarlos en una carpeta llamada Reportes.
- **Proceso en el Caso de Estudio:** Dado que no existe acceso directo a la base de datos original del sistema contable, este script permite recrear reportes mensuales basados en datos ingresados o simulados. Estos reportes son la base para el sistema de respaldo, ya que representan la información histórica que se desea preservar.

2. ActualizarReportes.py:

- **Funcionalidad:** Este script es responsable de actualizar el respaldo de la información contable en una base de datos secundaria y consolidada llamada BDDNuevoSistema.sqlite. Incluye una GUI con dos listas que muestran los archivos .csv en la carpeta Reportes y en una carpeta de respaldo, RespalDOS.
- **Rol en el Sistema:** Su rol principal es administrar el respaldo de los reportes mensuales, asegurándose de que todos los reportes nuevos sean copiados tanto a la base de datos BDDNuevoSistema como a la carpeta RespalDOS. Al seleccionar un archivo de la lista "Reportes" y presionar "Copiar", el sistema elimina registros duplicados en la base de datos según el mes y año del archivo seleccionado, luego importa los nuevos registros y respalda el archivo.
- **Proceso en el Caso de Estudio:** Este script asegura que la información contenida en los reportes mensuales de la Cooperativa se respalde de manera efectiva, manteniéndose accesible en la base de datos BDDNuevoSistema. De este modo, se garantiza que cada reporte mensual esté registrado y disponible en una única fuente consolidada de datos.

3. GenerarHistorico.py:

- **Funcionalidad:** Este script permite al usuario generar reportes históricos seleccionando un rango de fechas, exportando los registros

dentro del rango en un archivo .csv guardado en una carpeta llamada ReportesTemporales.

- **Rol en el Sistema:** Proporciona la capacidad de extraer información específica de la base de datos BDDNuevoSistema y generar reportes que cubran múltiples meses o años, cumpliendo así con la necesidad de un histórico accesible. El archivo generado tiene un nombre único (incluyendo un código hexadecimal de 5 dígitos), lo cual evita colisiones y facilita la organización de múltiples reportes históricos.
- **Proceso en el Caso de Estudio:** Este script permite a los usuarios de la Cooperativa acceder a reportes que incluyan todos los registros de un período específico. Esto es útil para análisis financieros o auditorías, ya que permite obtener una visión detallada del flujo de efectivo en el rango de fechas deseado.

Proceso de Respaldo de la Información

Para respaldar la información de la Cooperativa, el proceso se sigue de la siguiente manera:

1. Generación de Reportes:

- Los registros de ingresos y egresos se ingresan manualmente o se generan aleatoriamente usando SistemaContable.py. Estos registros son luego exportados como reportes mensuales en formato .csv a la carpeta Reportes.

2. Copia de Seguridad y Consolidación:

- Usando ActualizarReportes.py, los reportes en la carpeta Reportes se seleccionan y se copian a la carpeta RespalDOS. Además, los registros del archivo seleccionado se transfieren a la base de datos BDDNuevoSistema, eliminando cualquier dato duplicado del mismo mes y año. Esto asegura que la información de cada reporte mensual se consolide en una base de datos central sin redundancias.

3. Generación de Reportes Históricos:

- Cuando se necesita un análisis o reporte de un rango de fechas específico, GenerarHistorico.py permite seleccionar una fecha de inicio y fin. Luego, los registros dentro de ese rango se exportan a un archivo .csv en la carpeta ReportesTemporales, con un nombre único que

incluye un código hexadecimal de 5 dígitos. Este reporte temporal permite visualizar toda la información de un período, facilitando el análisis y la evaluación de datos históricos.

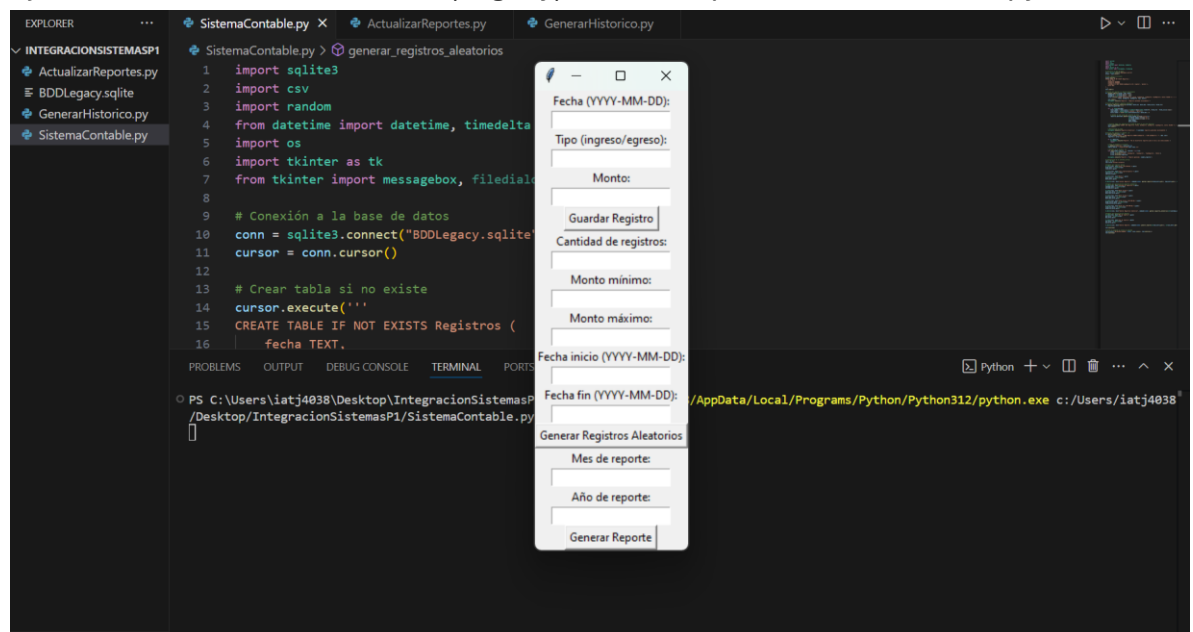
Validación y Control de Errores

El sistema también implementa varios mecanismos de validación y control de errores:

- **Formato de Fecha:** GenerarHistorico.py y ActualizarReportes.py validan las fechas de entrada, asegurando que el formato sea correcto y evitando errores de consulta o exportación.
- **Evitar Registros Duplicados:** Antes de insertar registros de un reporte mensual en BDDNuevoSistema, ActualizarReportes.py elimina cualquier registro existente del mismo mes y año, lo que evita la duplicación de datos en la base de datos consolidada.
- **Archivos CSV con Nombre Único:** GenerarHistorico.py utiliza un código hexadecimal único en el nombre de los archivos temporales, lo que evita que los reportes se sobrescriban accidentalmente.

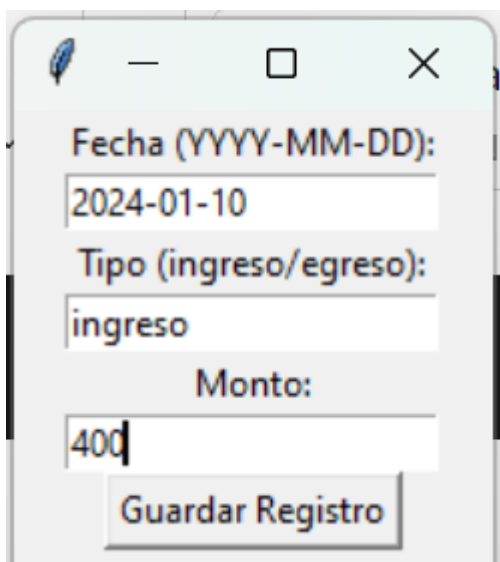
Evidencia de Implementación

1. Ejecución del sistema contable (Legacy) con el script “SistemaContable.py”:



Este sistema ofrece tres opciones:

1. En los 3 primeros campos, se puede crear manualmente un registro de ingreso/egreso que se almacenará en la base de datos legacy.



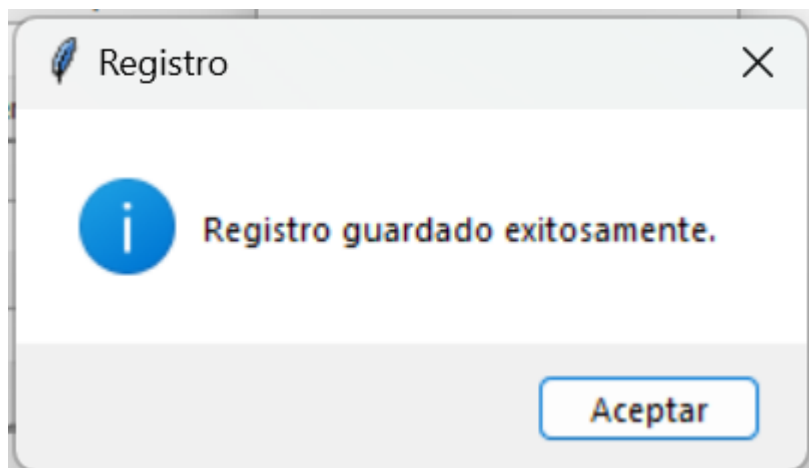
Formulario para crear un registro manual. Incluye campos para Fecha (YYYY-MM-DD), Tipo (ingreso/egreso) y Monto, con un botón Guardar Registro.

Fecha (YYYY-MM-DD):
2024-01-10

Tipo (ingreso/egreso):
ingreso

Monto:
400

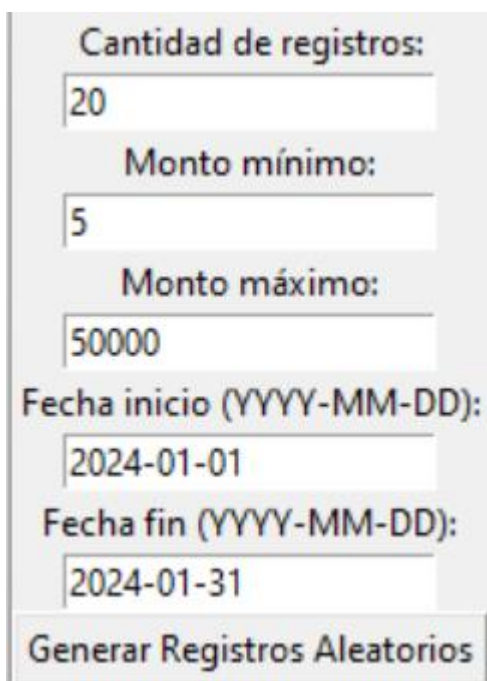
Guardar Registro



Registro guardado exitosamente.

Aceptar

2. En los campos del 4 al 8 permite ajustar parámetros para crear una cantidad “n” de registros con datos aleatorios dentro de un rango específico de fechas. Esto facilita la creación de datos para las pruebas del sistema.



Formulario para generar registros aleatorios. Incluye campos para Cantidad de registros, Monto mínimo, Monto máximo, Fecha inicio (YYYY-MM-DD) y Fecha fin (YYYY-MM-DD), con un botón Generar Registros Aleatorios.

Cantidad de registros:
20

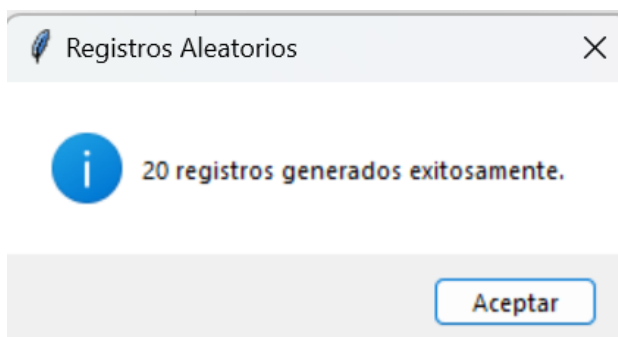
Monto mínimo:
5

Monto máximo:
50000

Fecha inicio (YYYY-MM-DD):
2024-01-01

Fecha fin (YYYY-MM-DD):
2024-01-31

Generar Registros Aleatorios



Registros Aleatorios

20 registros generados exitosamente.

Aceptar

3. Los últimos dos campos permiten seleccionar el mes y año del cual se quiere generar un reporte.

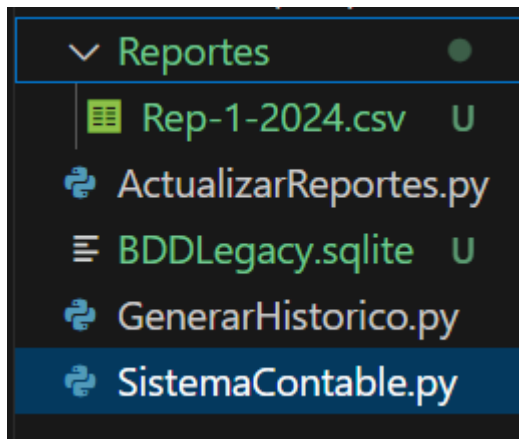
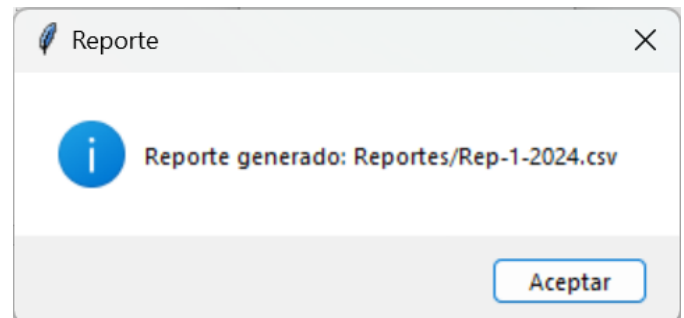
Mes de reporte:

01

Año de reporte:

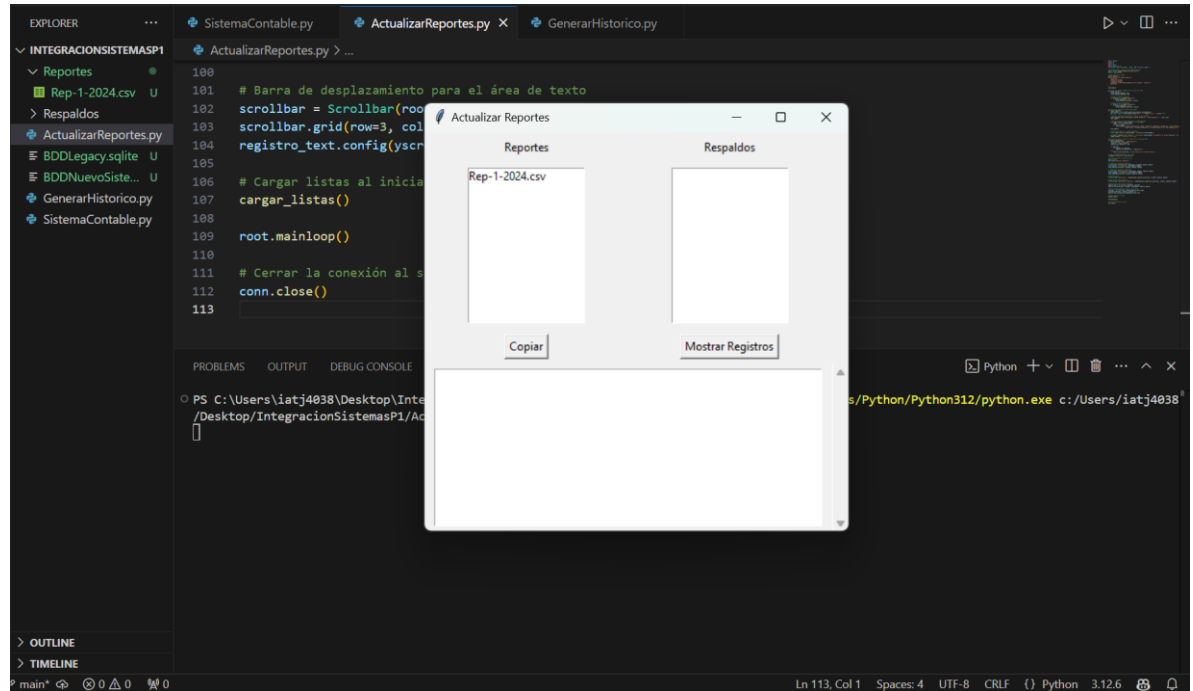
2024

Generar Reporte



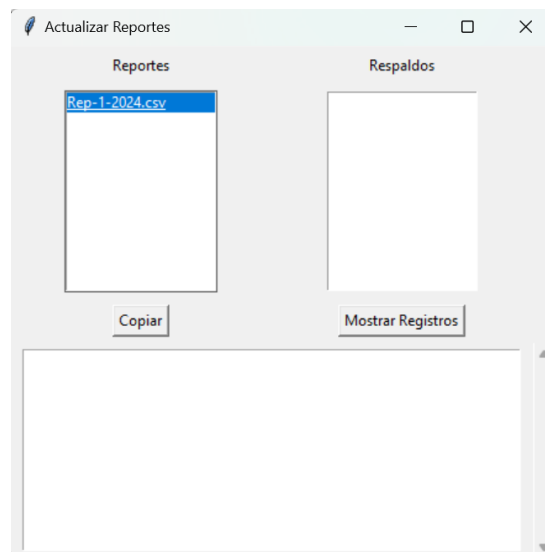
	A	B	C	D	E
1	Fecha,Mes,Reporte,AñoReporte,TipoRegistro,Monto				
2	2024-01-10,1,2024,ingreso,400.0				
3	2024-01-09,1,2024,egreso,5775.92				
4	2024-01-16,1,2024,ingreso,20870.34				
5	2024-01-18,1,2024,ingreso,31605.76				
6	2024-01-12,1,2024,egreso,18158.43				
7	2024-01-12,1,2024,ingreso,8605.64				
8	2024-01-13,1,2024,ingreso,29908.82				
9	2024-01-15,1,2024,ingreso,23436.07				
10	2024-01-25,1,2024,egreso,6556.78				
11	2024-01-22,1,2024,ingreso,8919.11				
12	2024-01-14,1,2024,egreso,21867.27				
13	2024-01-06,1,2024,egreso,24799.16				
14	2024-01-22,1,2024,egreso,17733.52				
15	2024-01-05,1,2024,ingreso,46772.49				
16	2024-01-26,1,2024,ingreso,17758.6				
17	2024-01-27,1,2024,ingreso,23606.66				
18	2024-01-26,1,2024,egreso,49125.21				
19	2024-01-27,1,2024,ingreso,17346.17				
20	2024-01-05,1,2024,egreso,8412.52				
21	2024-01-23,1,2024,egreso,24591.5				
22	2024-01-24,1,2024,ingreso,40604.37				
< >		Rep-1-2024		+	

2. Ejecución del Script “ActualizarReportes.py” para copiar los registros de un reporte generado por el sistema legacy en la nueva base de datos. El script evita registros duplicados en la nueva base al eliminar los registros que coincidan con las fechas del reporte antes de agregarlos nuevamente.

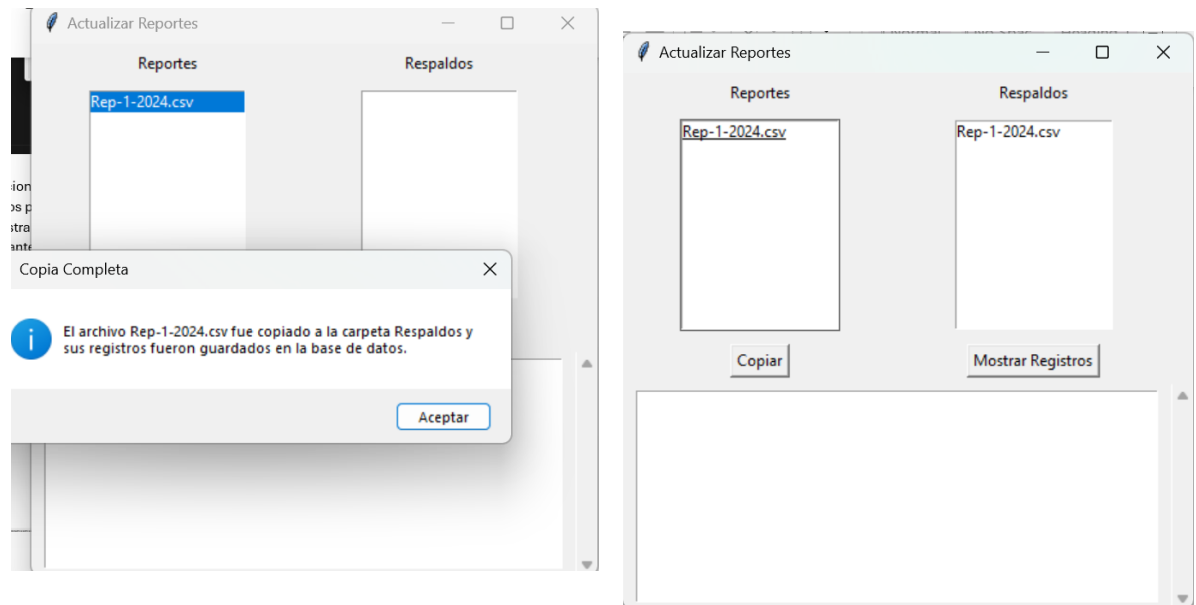


Este sistema permite seleccionar un reporte de la lista del lado izquierdo (todos los reportes generados por el legacy) y copiarlo en la nueva base de datos. También permite mostrar todos los registros de la nueva base de datos para validar que el proceso anterior se ejecutó correctamente.

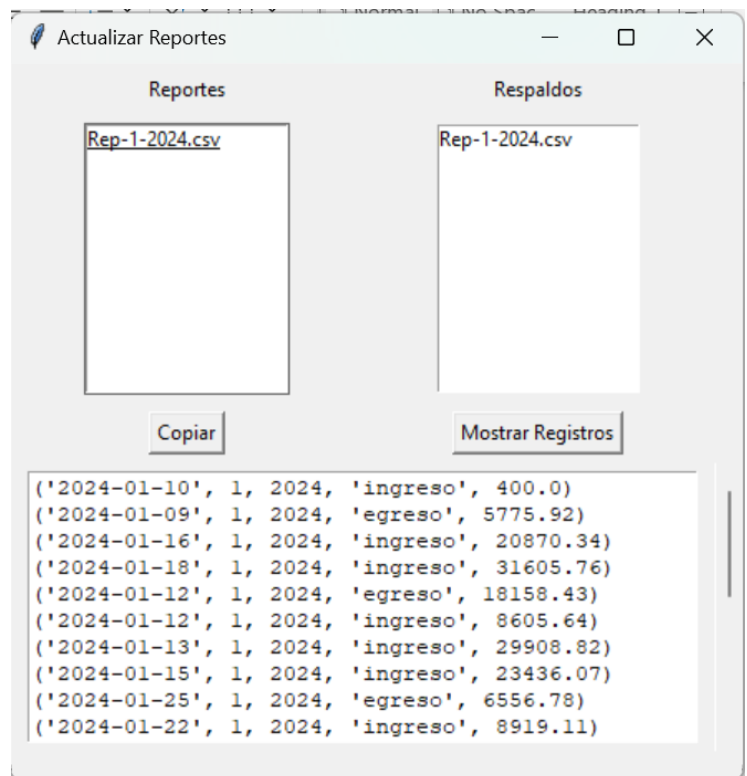
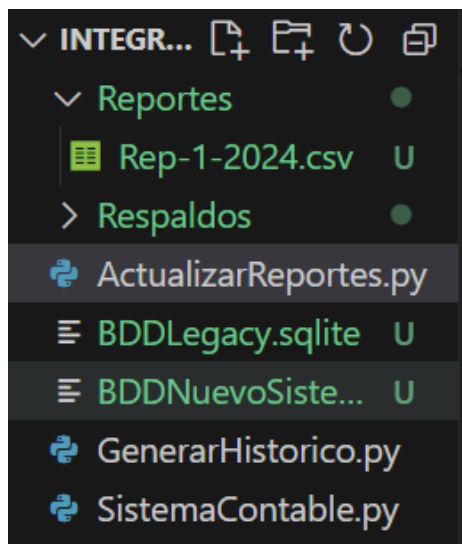
- a. Seleccionar y copiar reporte:



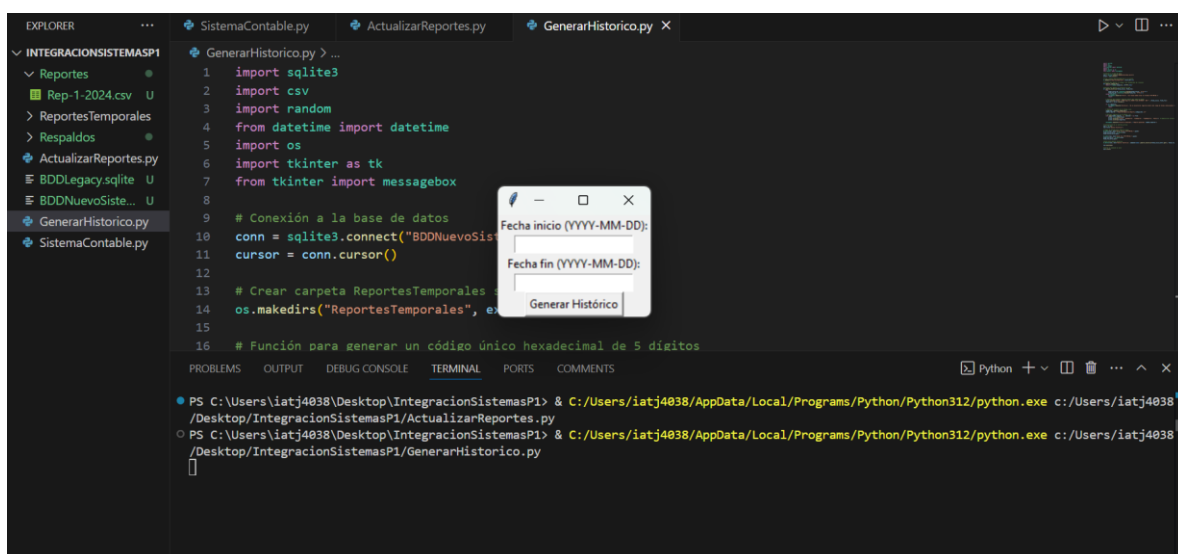
b. Pantalla que confirma el paso anterior:



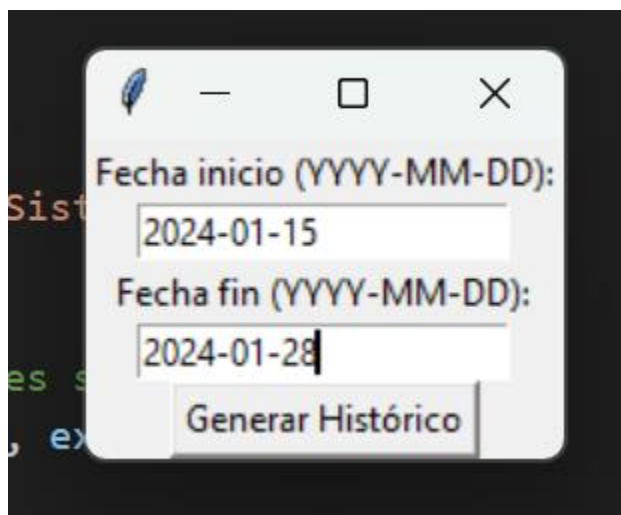
c. Mostrar los registros en la nueva base de datos “BDDNuevoSistema”:



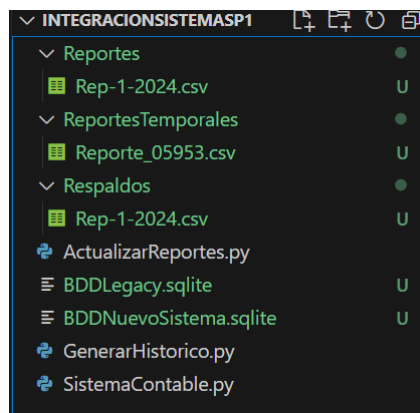
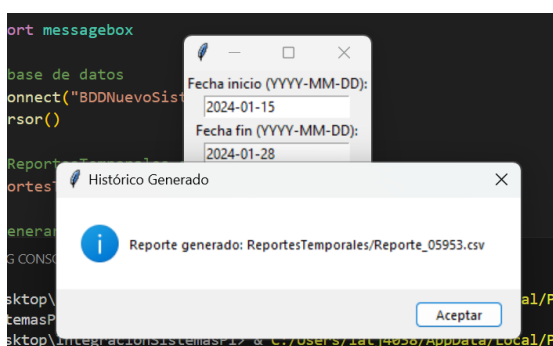
3. Ejecutar el script “GenerarHistorico.py”



- a. Ingresar las fechas de las que se quiere generar el reporte desde la nueva base de datos:



- b. Generar reporte de ese histórico:



c. Ver el reporte histórico generado desde la nueva base de datos:

INTEGRACIONSYSTEMASP1

Reportes

Rep-1-2024.csv

ReportesTemporales

Reporte_05953.csv

Respaldos

Rep-1-2024.csv

ActualizarReportes.py

BDDLegacy.sqlite

BDDNuevoSistema.sqlite

GenerarHistorico.py

SistemaContable.py

ReportesTemporales > Reporte_05953.csv

	Fecha,MesReporte,AñoReporte,TipoRegistro,Monto
1	
2	2024-01-16,1,2024,ingreso,20870.34
3	2024-01-18,1,2024,ingreso,31605.76
4	2024-01-15,1,2024,ingreso,23436.07
5	2024-01-25,1,2024,egreso,6556.78
6	2024-01-22,1,2024,ingreso,8919.11
7	2024-01-22,1,2024,egreso,17733.52
8	2024-01-26,1,2024,ingreso,17758.6
9	2024-01-27,1,2024,ingreso,23606.66
10	2024-01-26,1,2024,egreso,49125.21
11	2024-01-27,1,2024,ingreso,17346.17
12	2024-01-23,1,2024,egreso,24591.5
13	2024-01-24,1,2024,ingreso,40604.37
14	

Repositorio de la solución:

<https://github.com/Ivan-Tulcan/IntegracionP1>