

Universidad Gerardo Barrios

Facultad de Ciencia y Tecnología

Ingeniería en Sistemas y Redes Informáticas



ASIGNATURA: Programación Computacional III.

TEMA: Contenido a desarrollar (Desarrollo de proyecto Python.)

NOMBRE DE LA ACTIVIDAD: Laboratorio 1 – Semana 17.

PRESENTADO POR:

SMSS063523 – Argueta Portillo, Blanca Leticia.

SMSS084323 – Pérez Bonilla, Xavier Alexander.

SMSS098223 – Salgado Amaya, Jorge Alexis.

SMIS091121 – Zelaya Chávez, Kevin Alejandro.

DOCENTE:

Ing. WILLIAN MONTES.

CICLO VIII - 2024

San Miguel, noviembre 17 de 2024

Establecimiento de prioridades:

Priorizar las funcionalidades faltantes de tu Sistema de Administración y Ahorro de Dinero en Python, desglose de prioridades basado en las características pendientes:

Alta Prioridad:

Alertas de Gastos Hormiga:

Los gastos hormiga representan pequeñas transacciones que a menudo pasan desapercibidas, pero que, al acumularse, pueden comprometer el presupuesto del usuario. Al ser una funcionalidad enfocada en detectar y advertir sobre este tipo de gastos, proporciona un valor inmediato al usuario, mejorando su control financiero y evitando el desperdicio. Dada su relevancia en la prevención y monitoreo de este tipo de gastos, se prioriza para ser implementada de manera rápida.

Nota: “La decisión que tuvimos de usar la librería de Tkinter en lugar de continuar con PyQt5 se debe principalmente a razones de simplicidad y rapidez en el desarrollo. Tkinter es una biblioteca integrada de Python, lo que facilita su uso sin la necesidad de instalaciones adicionales. Además, es más ligera y rápida para construir interfaces gráficas básicas, como las requeridas en el proyecto, sin sacrificar funcionalidades esenciales.”

Desarrollo: Implementar lógica para identificar gastos repetidos y enviar alertas a través de Tkinter. Se puede usar threading para que las alertas no bloqueen la interfaz.

Establecimiento de Objetivos de Ahorro:

Permitir al usuario definir metas de ahorro es fundamental para promover una planificación financiera efectiva. Con objetivos claros, los usuarios pueden monitorear su progreso y ajustar sus hábitos de gasto para alcanzarlos, alineándose directamente con la misión principal de la aplicación: fomentar el ahorro consciente. Su relevancia en la generación de compromiso y motivación para el usuario lo convierte en una funcionalidad esencial.

Desarrollo: Agregar una interfaz para que el usuario establezca objetivos. Guardar y comparar el progreso contra estos objetivos utilizando SQLite.

Media Prioridad:

Notificaciones sobre Límites Presupuestarios:

Proporcionar advertencias cuando se alcanza un umbral específico de gasto ayuda a mantener un mejor control financiero.

Si bien este componente mejora el control financiero al alertar sobre la proximidad de los límites de gasto, su impacto no es tan inmediato ni crítico como el de las alertas de gastos hormiga o el establecimiento de objetivos. Proporciona una advertencia útil, pero no es indispensable para el funcionamiento inicial del sistema.

Desarrollo: Configurar valores límite en la base de datos para diferentes categorías y generar notificaciones cuando estos se alcancen.

Sugerencias de Ahorro Personalizado:

Ayuda a mejorar el manejo financiero, pero no es tan esencial como las alertas.

Ofrecer sugerencias basadas en el análisis de patrones de gasto puede ser muy valioso para el usuario, ya que proporciona recomendaciones para mejorar su manejo financiero. Sin embargo, esta funcionalidad actúa como una mejora o complemento, no como un requisito principal. Por ello, se prioriza después de funciones que tienen un impacto más directo.

Desarrollo: Implementar un análisis básico usando Pandas para ofrecer sugerencias basadas en patrones de gasto identificados.

Media Prioridad:

Simulador de Presupuesto:

Si bien el simulador de presupuesto ofrece un valor agregado al permitir que los usuarios prueben diferentes escenarios financieros, su impacto en el propósito principal del sistema es menos crucial que las funcionalidades críticas. Se puede abordar una vez que las funcionalidades de mayor prioridad están implementadas y estables.

Desarrollo: Crear un simulador que permita a los usuarios modificar ingresos y gastos para ver cómo afectarían su presupuesto general.

Resolución del plan de trabajo detallado para el segundo avance:

Comparación del Avance Entregado con el Plan de Trabajo Inicial.

Para asegurar que el proyecto esté alineado con el plan de trabajo trazado, se ha realizado una revisión comparativa del avance entregado con los objetivos planteados originalmente.

Plan de Trabajo Inicial:

Los objetivos iniciales del proyecto incluían las siguientes funcionalidades clave:

- Registro de ingresos y gastos.
- Categorización de los gastos por categorías predefinidas.
- Validación de las fechas para garantizar el registro correcto de los datos financieros.
- Visualización del progreso financiero mensual.
- Análisis detallado de los gastos mediante gráficos interactivos, representados por diagramas de pastel y gráficos de barras.
- Creación de una interfaz amigable para facilitar la entrada de datos y la interacción del usuario.

Segundo avance Entregado:

El avance realizado hasta la fecha abarca las siguientes funcionalidades:

- Registro de ingresos y gastos: Se ha implementado correctamente la función de agregar ingresos y gastos, con validación para el formato de la fecha. Esto asegura que los datos ingresados sean precisos, cumpliendo con el objetivo establecido en el plan inicial.
- Categorización de gastos: Los gastos se pueden clasificar mediante un conjunto de categorías predefinidas, lo que simplifica la organización de las transacciones y facilita el análisis financiero.
- Visualización del progreso financiero: La funcionalidad que permite mostrar el progreso mensual de los ingresos y gastos ha sido desarrollada utilizando

gráficos interactivos mediante la biblioteca Plotly, lo cual también estaba contemplado en los objetivos iniciales.

- **Análisis de gastos:** Se ha implementado la visualización de los gastos mediante gráficos, permitiendo un análisis detallado y visual de los mismos. Esto se realiza mediante gráficos de pastel para mostrar la distribución de gastos por categoría y gráficos de barras para observar el progreso mensual.
- **Interfaz amigable:** Se ha desarrollado una interfaz utilizando Tkinter, proporcionando una experiencia sencilla para el ingreso de datos, tal como se planteó.

El avance entregado hasta el momento se ha mantenido fiel con los objetivos propuestos en el plan de trabajo inicial. Todas las funcionalidades propuestas han sido desarrolladas, asegurando la coherencia del proyecto y el cumplimiento de los plazos establecidos. La elección de utilizar Tkinter para la interfaz ha permitido mantener una experiencia accesible y amigable para el usuario, en línea con los objetivos definidos desde el inicio del proyecto.

Objetivos faltantes y Plan de desarrollo:

Funcionalidades y Componentes Planificados para el Tercer Avance y la Entrega Final.

1. Implementación de Alertas de Gastos Hormiga

Se implementará un sistema que notifique al usuario cuando los gastos menores acumulados (gastos hormiga) superen un límite predefinido.

Plan de Desarrollo:

Usar funciones que monitoreen los gastos menores registrados por categoría y acumulen su valor.

Establecer un umbral configurable por el usuario para activar la alerta.

Utilizar mensajes emergentes de tkinter.messagebox para mostrar las alertas en tiempo real.

Estado Actual: No implementado.

Objetivo: Proporcionar control inmediato sobre los pequeños gastos que podrían pasar desapercibidos.

2. Sistema de Alertas y Notificaciones Personalizadas

Crear un módulo que permita a los usuarios configurar notificaciones relacionadas con la superación de sus límites de gasto por categoría o presupuesto general.

Plan de Desarrollo:

Ampliar la lógica de categorización actual para monitorear los gastos en tiempo real. Añadir opciones en la interfaz para que el usuario pueda definir límites.

Usar notificaciones emergentes con tkinter o integración con librerías externas si es necesario para notificaciones avanzadas.

Estado Actual: No implementado.

Objetivo: Facilitar un control financiero más efectivo, alertando al usuario sobre el riesgo de exceder sus presupuestos.

3. Simulador de Presupuesto y Objetivos de Ahorro

Desarrollar un módulo de simulación de presupuesto que permita al usuario planificar objetivos de ahorro y simular escenarios futuros basados en sus ingresos y gastos actuales.

Plan de Desarrollo:

Utilizar cálculos matemáticos y estadísticas para simular proyecciones de ahorro según metas definidas por el usuario.

Implementar gráficos interactivos con plotly para mostrar las proyecciones. Proporcionar un cuadro de diálogo para ingresar objetivos de ahorro y parámetros de simulación.

Estado Actual: No implementado.

Objetivo: Ayudar al usuario a planificar y visualizar escenarios futuros, promoviendo una mejor toma de decisiones financieras.

4. Optimización de la Base de Datos

Mejorar el manejo y consultas a la base de datos SQLite para optimizar el rendimiento, sobre todo en aplicaciones que manejan grandes volúmenes de datos.

Plan de Desarrollo:

Revisar y mejorar la estructura de las tablas y el índice de consultas.
Implementar mecanismos de cacheo si es necesario para agilizar el acceso a los datos.

Asegurar la integridad de los datos y evitar bloqueos en consultas concurrentes.

Estado Actual: Implementación inicial de base de datos funcional.

Objetivo: Asegurar un rendimiento óptimo y una experiencia fluida para el usuario.

5. Mejoras en la Interfaz de Usuario

Realizar ajustes y mejoras en la interfaz de Tkinter para proporcionar una experiencia más amigable y visualmente atractiva.

Plan de Desarrollo:

Rediseñar algunos elementos de la interfaz para mayor claridad.
Utilizar componentes adicionales de tkinter como ttk para mejorar la apariencia.
Agregar menús contextuales y atajos de teclado.

Estado Actual: Interfaz funcional con componentes básicos.

Objetivo: Facilitar la navegación y el uso del sistema, aumentando la usabilidad y satisfacción del usuario.

Plan General de Desarrollo y Ajustes Respecto a Entregas Anteriores.

Uso de threading: Se continuará utilizando threading para evitar que la interfaz se congele durante la generación de gráficos o consultas a la base de datos.

Plotly y Pandas: Estas bibliotecas seguirán siendo clave para la creación de gráficos interactivos y el manejo de datos. Se evaluará la necesidad de mejorar las visualizaciones según el feedback recibido.

Gestión de Errores y Validaciones Adicionales: Se agregaron validaciones para asegurar que los datos ingresados son correctos y para manejar posibles fallos del sistema de forma más robusta.

Con estas funcionalidades, el sistema quedará completamente implementado para cubrir las necesidades de gestión financiera planteadas inicialmente, proporcionando un control más efectivo y una visión integral del manejo de ingresos, gastos y objetivos financieros de los usuarios que la usen.