Sistema de Registro de Gastos Personales

Universidad del Valle de México

Gadiel Iván Núñez Gutiérrez

17 de junio de 2025

# Índice

Tabla de contenido

[Índice 2](#_Toc201050480)

[Índice de imágenes 2](#_Toc201050481)

[Capítulo 1 Definición 3](#_Toc201050482)

[1.1 Definición del problema 3](#_Toc201050483)

[1.2 Objetivos 3](#_Toc201050484)

[Capítulo 2 Estado del arte 4](#_Toc201050485)

[Capítulo 3 Marco teórico 5](#_Toc201050486)

[3.1 Bases de datos relacionales y SQLite 5](#_Toc201050487)

[3.2 Lenguaje Python y programación orientada a eventos 5](#_Toc201050488)

[3.3 Interfaces gráficas (GUI) 5](#_Toc201050489)

[Capítulo 4 Diseño 6](#_Toc201050490)

[Imagen 1. Diagrama de Casos de Uso 6](#_Toc201050491)

[Imagen 2. Diagrama Entidad-Relación 6](#_Toc201050492)

[Capítulo 5. Desarrollo 7](#_Toc201050493)

[Imagen 3. Interfaz gráfica del sistema en ejecución 9](#_Toc201050494)

[Conclusión 9](#_Toc201050495)

[Fuentes de información 10](#_Toc201050496)

# Índice de imágenes

Imagen 1. Diagrama de casos de uso ...................................... 6

Imagen 2. Diagrama Entidad-Relación .................................... 6

Imagen 3. Interfaz gráfica del sistema en ejecución…………..10

# Capítulo 1 Definición

## 1.1 Definición del problema

Actualmente, muchas personas no llevan un control adecuado de sus gastos personales, lo que provoca falta de organización financiera. Este proyecto propone una aplicación sencilla que permita registrar, consultar y eliminar gastos personales de forma eficiente y visual.

## 1.2 Objetivos

1.2.1 Objetivo General

Desarrollar una aplicación de escritorio para el registro de gastos personales que permita al usuario llevar un control simple y efectivo.

1.2.2 Objetivos Específicos

- Crear una interfaz intuitiva para ingresar gastos.

- Permitir al usuario consultar y eliminar sus registros de gastos.

- Utilizar una base de datos para almacenar la información.

# Capítulo 2 Estado del arte

Existen diversas aplicaciones móviles y de escritorio dedicadas a la gestión de gastos, como Fintonic, Monefy o Wallet. Estas aplicaciones ofrecen funcionalidades avanzadas, pero muchas veces resultan innecesarias para usuarios que solo desean registrar gastos simples. Este proyecto busca simplificar el proceso sin requerir conexión a internet o funciones complejas.

# Capítulo 3 Marco teórico

Un sistema de registro de gastos personales permite al usuario almacenar información sobre sus egresos en una base de datos para su posterior consulta y análisis. Este tipo de sistemas se basa en conceptos de bases de datos relacionales, interfaces gráficas y programación orientada a eventos. Utilizaremos SQLite por su ligereza y facilidad de integración con Python.

## 3.1 Bases de datos relacionales y SQLite

Una **base de datos relacional** permite organizar la información en forma de tablas relacionadas entre sí mediante claves primarias y foráneas. En este proyecto se utiliza **SQLite**, un sistema de gestión de bases de datos ligero, embebido y que no requiere servidor. Esto lo convierte en una opción ideal para aplicaciones personales y de escritorio.

Al ser una base de datos local, los datos se almacenan directamente en un archivo .db, lo que permite su portabilidad sin necesidad de instalaciones complejas. Además, SQLite es muy fácil de conectar con lenguajes como **Python**, lo cual facilita el desarrollo rápido.

## 3.2 Lenguaje Python y programación orientada a eventos

**Python** es un lenguaje de programación interpretado, conocido por su simplicidad y legibilidad. Permite desarrollar desde scripts pequeños hasta aplicaciones completas. Es compatible con múltiples bibliotecas para desarrollo de aplicaciones de escritorio, web, ciencia de datos, entre otros.

En este proyecto se utiliza **Tkinter**, una biblioteca estándar de Python que permite crear interfaces gráficas (ventanas, botones, cuadros de texto, etc.). El funcionamiento del sistema se basa en **eventos**, es decir, reacciones a acciones del usuario como hacer clic en un botón o escribir un dato.

## 3.3 Interfaces gráficas (GUI)

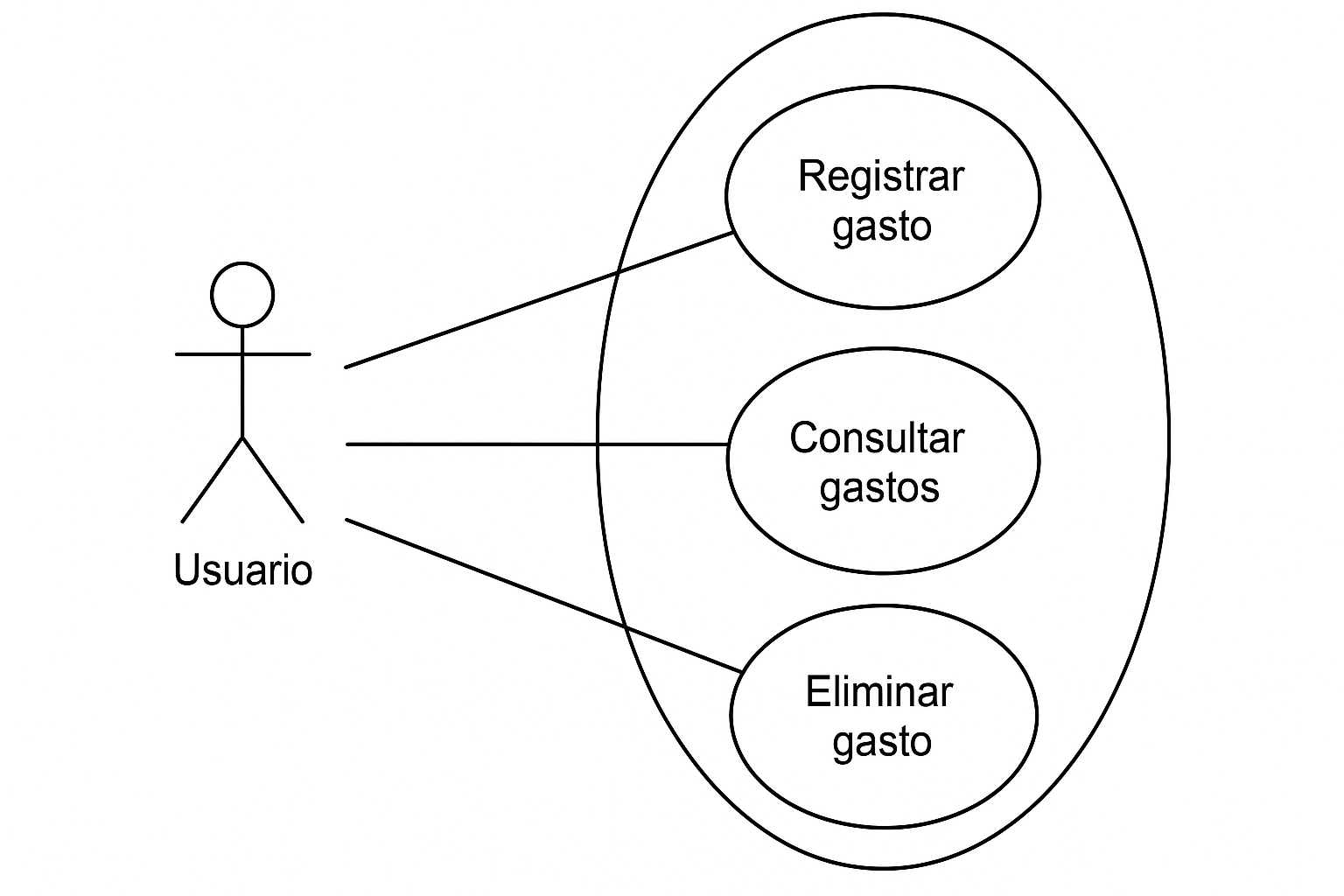
Una **interfaz gráfica de usuario (GUI)** permite que las personas interactúen con el sistema sin necesidad de usar comandos de consola. Los botones, menús, cuadros de texto y ventanas hacen que la experiencia sea más amigable, sobre todo para usuarios que no están familiarizados con programación.

El sistema desarrollado permite a cualquier persona registrar un gasto escribiendo su categoría, cantidad y fecha. Luego, con un solo clic, esta información se guarda en la base de datos.

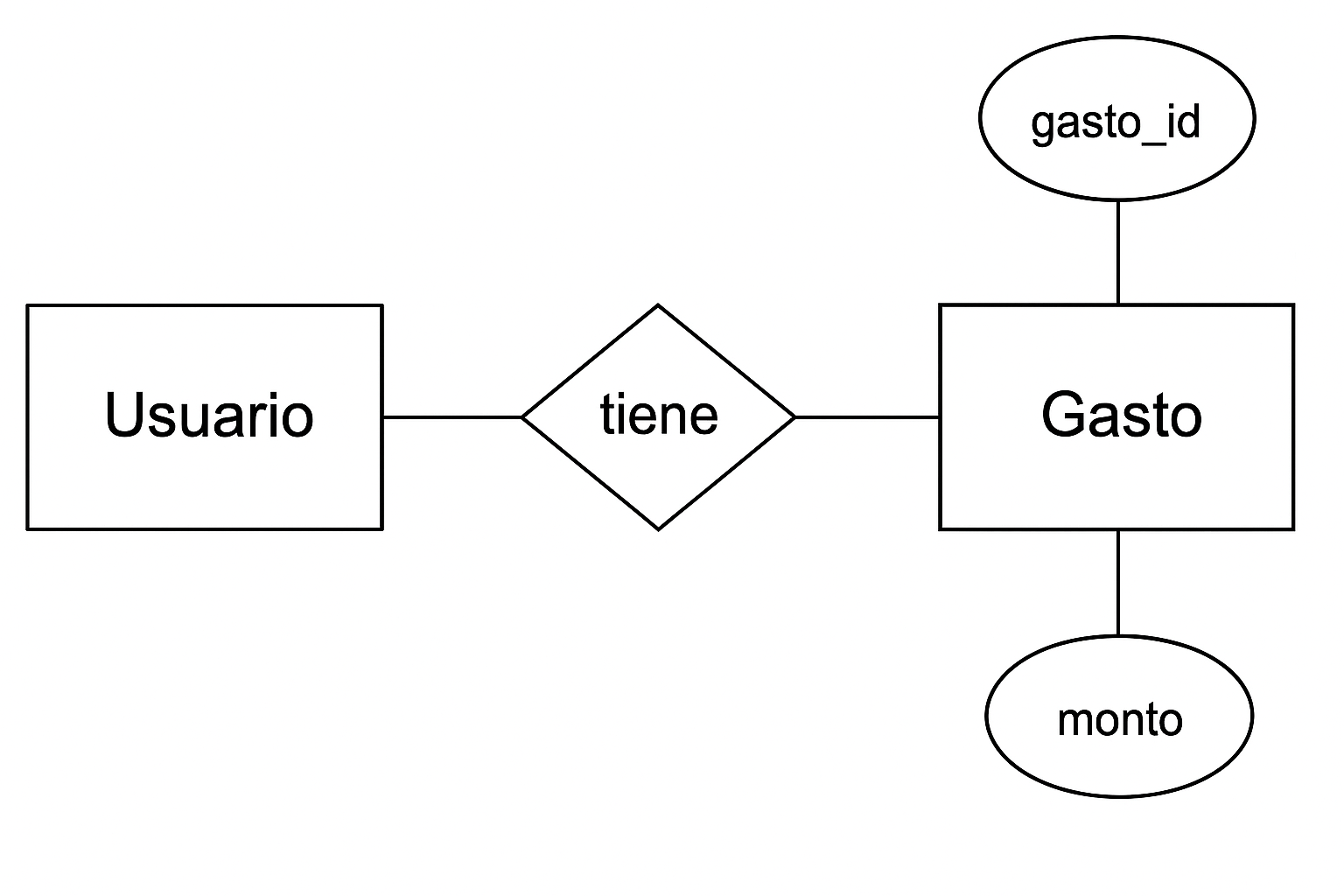
# Capítulo 4 Diseño

A continuación, se muestran los diagramas del sistema propuesto:

## Imagen 1. Diagrama de Casos de Uso



## Imagen 2. Diagrama Entidad-Relación



# Capítulo 5. Desarrollo

**Herramientas utilizadas**

* Lenguaje: Python 3
* Librerías: sqlite3, tkinter
* IDE: Visual Studio Code

**Algoritmos y lógica**

* Conexión a SQLite
* Inserción y consulta de datos
* Interfaz gráfica con botones y cuadros de texto

**Código principal**

python

CopiarEditar

import sqlite3

from tkinter import \*

def crear\_db():

conn = sqlite3.connect('gastos.db')

cursor = conn.cursor()

cursor.execute('''CREATE TABLE IF NOT EXISTS gastos (

id INTEGER PRIMARY KEY AUTOINCREMENT,

categoria TEXT NOT NULL,

cantidad REAL NOT NULL,

fecha TEXT NOT NULL

)''')

conn.commit()

conn.close()

def insertar\_gasto():

conn = sqlite3.connect('gastos.db')

cursor = conn.cursor()

cursor.execute("INSERT INTO gastos (categoria, cantidad, fecha) VALUES (?, ?, ?)",

(entrada\_categoria.get(), entrada\_cantidad.get(), entrada\_fecha.get()))

conn.commit()

conn.close()

entrada\_categoria.delete(0, END)

entrada\_cantidad.delete(0, END)

entrada\_fecha.delete(0, END)

app = Tk()

app.title("Control de gastos")

app.geometry("300x200")

Label(app, text="Categoría:").pack()

entrada\_categoria = Entry(app)

entrada\_categoria.pack()

Label(app, text="Cantidad:").pack()

entrada\_cantidad = Entry(app)

entrada\_cantidad.pack()

Label(app, text="Fecha (YYYY-MM-DD):").pack()

entrada\_fecha = Entry(app)

entrada\_fecha.pack()

Button(app, text="Agregar gasto", command=insertar\_gasto).pack()

crear\_db()

app.mainloop()

**Pantalla del sistema funcionando**

## Imagen 3. Interfaz gráfica del sistema en ejecución Captura de pantalla de computadora El contenido generado por IA puede ser incorrecto.

# 

# Conclusión

El sistema desarrollado cumple con el objetivo principal de permitir la gestión local de gastos personales de forma sencilla. Se logró implementar una base de datos funcional, una interfaz amigable y funciones básicas de inserción y consulta. Además, al estar construido con tecnologías libres, puede ser mejorado fácilmente.

# 

# Fuentes de información

* Python Software Foundation. (2025). *Python documentation*. <https://docs.python.org>
* SQLite Consortium. (2025). *SQLite documentation*. <https://www.sqlite.org>
* TkDocs. (2025). *Tkinter documentation*. <https://tkdocs.com/tutorial>