



# LIBRERIAS EN PYTHON

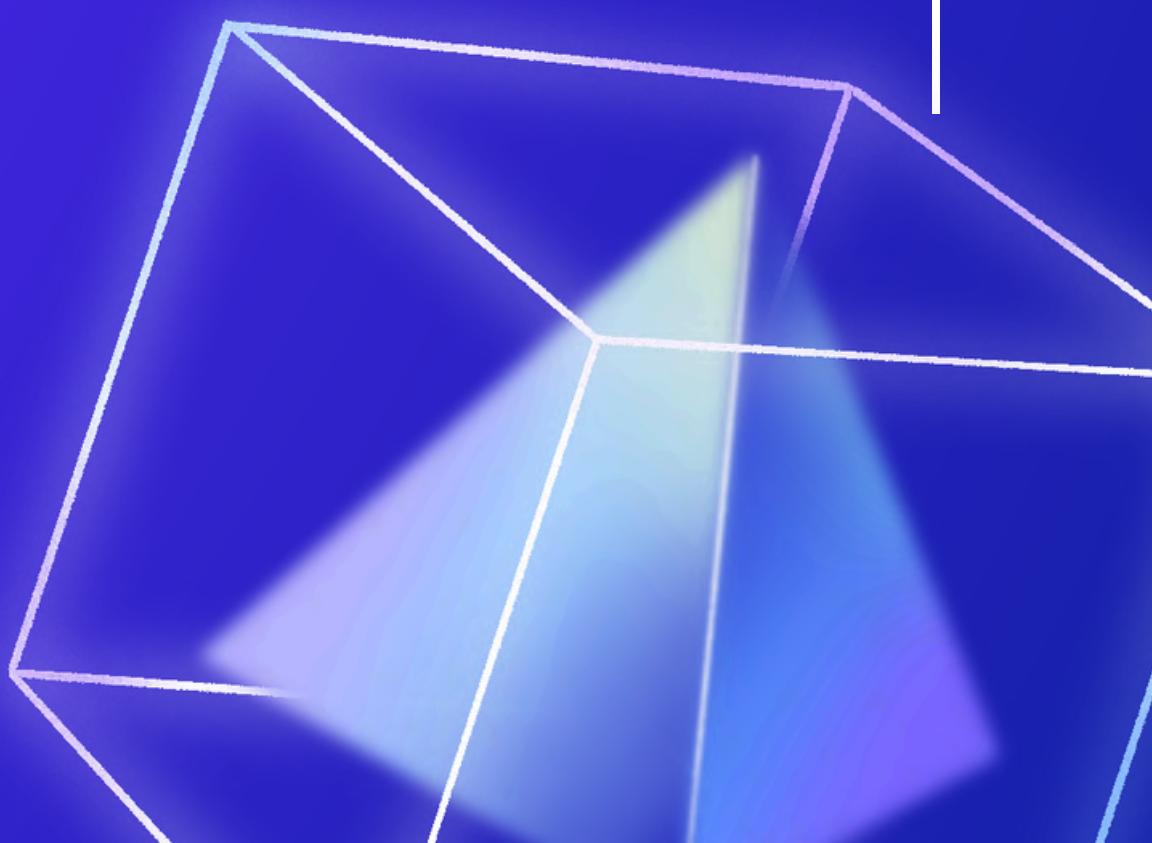
# DATETIME

Daniel Alejandro Castro Becerra  
20242020271



# TABLA DE CONTENIDOS

- |   |    |
|---|----|
| • ¿Qué es datetime?                       | 01 |
| • Componentes principales                 | 02 |
| • Ejemplos prácticos                      | 03 |
| • Importancia en sistemas reales          | 04 |
| • Buenas prácticas y problemas<br>comunes | 05 |



# ¿QUÉ ES DATETIME?

Es una librería estándar de Python que permite trabajar con fechas y horas.

Se usa para:

- Registrar eventos
  - Calcular diferencias de tiempo
  - Programar tareas
  - Analizar datos con marcas de tiempo
  - Automatización
- 



# COMPONENTES PRINCIPALES



## DATE

- Representa únicamente la fecha, sin incluir información de hora.
- Contiene año, mes y día.
- Se usa cuando solo te interesa trabajar con días específicos y no con horas



## TIME

- Solo la hora, minutos, segundos y opcionalmente microsegundos.
- No tiene información de fecha.
- Útil para representar horarios, turnos, cronómetros o alarmas.



## DATETIME

- Combina fecha y hora en un solo objeto.
- Es el más usado porque permite manejar eventos con fecha y hora exacta.
- Se puede comparar, sumar o restar con otros datetime.



## TIMDELTA

- Permite calcular diferencias entre fechas o tiempos, o sumar/restar periodos.
- Útil para saber cuánto falta para un evento, cuántos días han pasado, etc.



## STRFTIME()

- Significa "string format time".
- Convierte un objeto date, time o datetime en texto con el formato que quieras.
- Muy útil para mostrar fechas en pantallas, reportes o logs.



## STRPTIME()

- Significa "string parse time".
- Hace lo contrario de strftime(): convierte texto que representa una fecha/hora en un objeto datetime.
- Útil cuando recibes fechas como texto y quieres manipularlas.

# EJEMPLOS PRACTICOS

```
from datetime import datetime  
  
momento_actual = datetime.now()  
print("El usuario inició sesión en:", momento_actual)
```

El usuario inició sesión en: 2026-02-11 08:21:53.674824

## REGISTRO DE ACCESO A UN SISTEMA

- Apps bancarias
- Plataformas educativas
- Sistemas empresariales
- Historial de actividad

```
from datetime import datetime  
  
fecha_vencimiento = datetime(2026, 3, 1)  
hoy = datetime.now()  
  
diferencia = fecha_vencimiento - hoy  
  
print("Faltan", diferencia.days, "días para el vencimiento.")
```

Faltan 17 días para el vencimiento.

## CALCULAR DÍAS RESTANTES PARA UN VENCIMIENTO

- Bancos
- Pagos de servicios
- Recordatorios automáticos
- Cálculo de intereses

```
from datetime import datetime  
  
fecha_texto = "11-02-2026"  
fecha_convertida = datetime.strptime(fecha_texto, "%d-%m-%Y")  
  
print(fecha_convertida.strftime("%Y-%m-%d"))
```

2026-02-11

## CONVERTIR TEXTO A FECHA

- Formularios web
- Archivos CSV
- Sistemas contables
- Importación de datos

# ¿POR QUÉ ES IMPORTANTE EN SISTEMAS REALES?

Porque todos los sistemas registran eventos que son medidos en el tiempo.

01

04

03

02

## PRECISIÓN EN TRANSACCIONES FINANCIERAS

los sistemas bancarios y comerciales dependen de fechas exactas para registrar pagos, cobros y movimientos de manera confiable.

## EVITAR PÉRDIDAS ECONÓMICAS

un error en la fecha de un registro puede generar inconsistencias, retrasos o incluso pérdidas financieras.

## AUTOMATIZACIÓN BASADA EN TIEMPO

muchos procesos automáticos, como envíos de correos, reportes o sistemas de control, dependen de cálculos de tiempo precisos.



# PROBLEMAS COMUNES Y BUENAS PRACTICAS



- Diferencias entre zonas horarias (UTC vs hora local) que pueden afectar registros de eventos simultáneos.
- Manejo de años bisiestos y fechas especiales que pueden generar errores de cálculo.
- Confusión entre distintos formatos de fecha (DD/MM/AAAA vs MM/DD/AAAA).
- Falta de validación de datos de entrada, lo que puede provocar registros incorrectos o inconsistentes.



- Implementar conversiones consistentes entre zonas horarias.
- Considerar reglas de años bisiestos en cálculos de fechas.
- Definir un formato estándar para todas las fechas del sistema.
- Validar siempre los datos de entrada antes de procesarlos para evitar errores.

# GRACIAS!

## REPOSITORIO:

[https://github.com/daneaaal/biblioteca-datetime-python.](https://github.com/daneaaal/biblioteca-datetime-python)

Aprende con Alf. (s.f.). La librería Datetime [Tutorial en línea]. Recuperado de <https://aprendeconalf.es/docencia/python/manual/datetime/>

Python Software Foundation. (s.f.). datetime – Tipos básicos de fecha y hora. En Documentación oficial de Python. Recuperado de <https://docs.python.org/es/3/library/datetime.html>

Ortiz Ordoñez, J. (2019, 13 de septiembre). Python – Ejercicio 3: Obtener la Fecha y Hora Actuales del Sistema con el Módulo datetime. YouTube. [https://www.youtube.com/watch?v=WxQnV\\_aIXaE](https://www.youtube.com/watch?v=WxQnV_aIXaE)

