



## UNIVERSIDAD DISTRITAL FJDC

Facultad de Ingeniería

---

# Biblioteca datetime en Python

---

**Asignatura:** Probabilidad y Estadística

**Docente:** Alberto Acosta Lopez

**Estudiante:** Daniel Alejandro Castro Becerra

**Código:** 20242020271

# Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Componentes Principales</b>	<b>2</b>
2.1. Clase date . . . . .	2
2.2. Clase time . . . . .	2
2.3. Clase datetime . . . . .	2
2.4. Clase timedelta . . . . .	3
<b>3. Funciones Principales</b>	<b>3</b>
3.1. datetime.now() . . . . .	3
3.2. strftime() . . . . .	3
3.3. strptime() . . . . .	3
<b>4. Ejemplo Práctico Completo</b>	<b>3</b>
<b>5. Importancia en Sistemas Reales</b>	<b>4</b>
<b>6. Problemas Comunes</b>	<b>4</b>
<b>7. Buenas Prácticas</b>	<b>4</b>
<b>8. Conclusiones</b>	<b>5</b>
<b>9. Bibliografía</b>	<b>5</b>

# 1. Introducción

La biblioteca `datetime` es un módulo estándar de Python que permite manipular fechas y horas de manera eficiente. Es una herramienta fundamental en el desarrollo de software moderno, ya que prácticamente todos los sistemas informáticos dependen del manejo correcto del tiempo.

Desde sistemas bancarios hasta plataformas educativas, el control preciso de fechas y horas garantiza la integridad de los datos y la correcta ejecución de procesos automáticos.

## 2. Componentes Principales

### 2.1. Clase date

Representa una fecha específica compuesta por año, mes y día. Se utiliza cuando no es necesario trabajar con información de hora.

Ejemplo:

```
from datetime import date

fecha = date(2026, 5, 20)
print(fecha)
```

### 2.2. Clase time

Representa únicamente la hora (hora, minutos, segundos y microsegundos).

```
from datetime import time

hora = time(14, 30, 0)
print(hora)
```

### 2.3. Clase datetime

Combina fecha y hora en un solo objeto. Es la clase más utilizada.

```
from datetime import datetime

ahora = datetime.now()
print(ahora)
```

## 2.4. Clase timedelta

Permite realizar operaciones matemáticas con fechas y horas.

```
from datetime import datetime, timedelta

hoy = datetime.now()
manana = hoy + timedelta(days=1)

print("Hoy:", hoy)
print("Mañana:", manana)
```

## 3. Funciones Principales

### 3.1. datetime.now()

Devuelve la fecha y hora actual del sistema.

### 3.2. strftime()

Convierte un objeto datetime en una cadena de texto con formato específico.

```
from datetime import datetime

ahora = datetime.now()
formato = ahora.strftime("%d/%m/%Y %H:%M")
print(formato)
```

### 3.3. strptime()

Convierte una cadena de texto en un objeto datetime.

```
from datetime import datetime

texto = "25/12/2026"
fecha = datetime.strptime(texto, "%d/%m/%Y")
print(fecha)
```

## 4. Ejemplo Práctico Completo

Supongamos que queremos calcular cuántos días faltan para finalizar el año.

```

from datetime import datetime

hoy = datetime.now()
fin_ano = datetime(hoy.year, 12, 31)

diferencia = fin_ano - hoy

print("Hoy es:", hoy.strftime("%d/%m/%Y"))
print("Faltan", diferencia.days, "días para finalizar el año.")

```

Este tipo de cálculo es común en aplicaciones financieras, sistemas de planificación y gestión empresarial.

## 5. Importancia en Sistemas Reales

El manejo adecuado del tiempo es esencial en:

- Sistemas bancarios para registrar pagos e intereses.
- Plataformas educativas para fechas de entrega.
- Sistemas empresariales para auditoría y control.
- Automatización de procesos programados.

Un error en el manejo de fechas puede generar pérdidas económicas, inconsistencias de datos o fallos en procesos críticos.

## 6. Problemas Comunes

- Confusión entre formatos de fecha (DD/MM/AAAA vs MM/DD/AAAA).
- Diferencias de zonas horarias.
- Manejo incorrecto de años bisiestos.
- Falta de validación en datos ingresados por el usuario.

## 7. Buenas Prácticas

- Definir un formato estándar para fechas.
- Validar datos de entrada.

- Considerar el uso de UTC en sistemas distribuidos.
- Documentar el formato utilizado en bases de datos.

## 8. Conclusiones

La biblioteca datetime es una herramienta esencial en el desarrollo de software. Permite gestionar fechas y horas de manera precisa, facilitando la creación de sistemas confiables y automatizados.

Su correcta implementación reduce errores, mejora la integridad de los datos y garantiza el funcionamiento adecuado de aplicaciones críticas.

## 9. Bibliografía

- Python Software Foundation. (s. f.). *datetime - Basic date and time types*. Python Documentation. Disponible en: <https://docs.python.org/3/library/datetime.html>
- Aprende con Alf. (s. f.). *La libreria datetime*. Disponible en: <https://aprendeconalf.es/docencia/python-basico/librerias/datetime/>
- Ortiz Ordóñez, J. (2019, septiembre 13). *Python - Ejercicio 3: Obtener la fecha y hora actuales del sistema con el modulo datetime* [Video]. YouTube. Disponible en: [https://www.youtube.com/watch?v=WxQrV\\_aIXaE](https://www.youtube.com/watch?v=WxQrV_aIXaE)