El código anterior ya tiene una estructura de herencia de tres niveles: la clase base Cita y las clases derivadas CitaMedicoGeneral , CitaOdontologica y CitaGinecologica . Cada una de estas clases ya tiene varios métodos definidos.

En el código proporcionado, se puede ver un ejemplo práctico de sobreescritura de métodos en las clases CitaMedicoGeneral , CitaOdontologica y CitaGinecologica . Todas estas clases heredan de la clase base Cita y sobrescriben el método obtener\_informacion() para proporcionar detalles específicos sobre la cita según el tipo de cita.

Aquí está el ejemplo de sobreescritura del método en la clase CitaMedicoGeneral :

class CitaMedicoGeneral(Cita):

def \_\_init\_\_(self, fecha, hora, paciente, medico):

super().\_\_init\_\_(fecha, hora, paciente)

self.medico = medico

def obtener\_informacion(self):

# Lógica para obtener información de la cita con médico general

print(f"La cita con el médico general {self.medico} está programada para el día {self.fecha} a las {self.hora}. ")

En este código, obtener\_informacion() se sobrescribe para imprimir un mensaje que incluye el nombre del médico, la fecha y la hora de la cita. Este mensaje es específico para las citas con el médico general. Las clases CitaOdontologica y CitaGinecologica tienen una lógica similar para sobrescribir el método obtener\_informacion() .

En resumen, la sobreescritura de métodos es una característica de la programación orientada a objetos que permite a una subclase proporcionar una implementación diferente de un método que ya está definido en su superclase. Esto es útil cuando la subclase necesita cambiar el comportamiento del método para adaptarse a sus necesidades específicas.