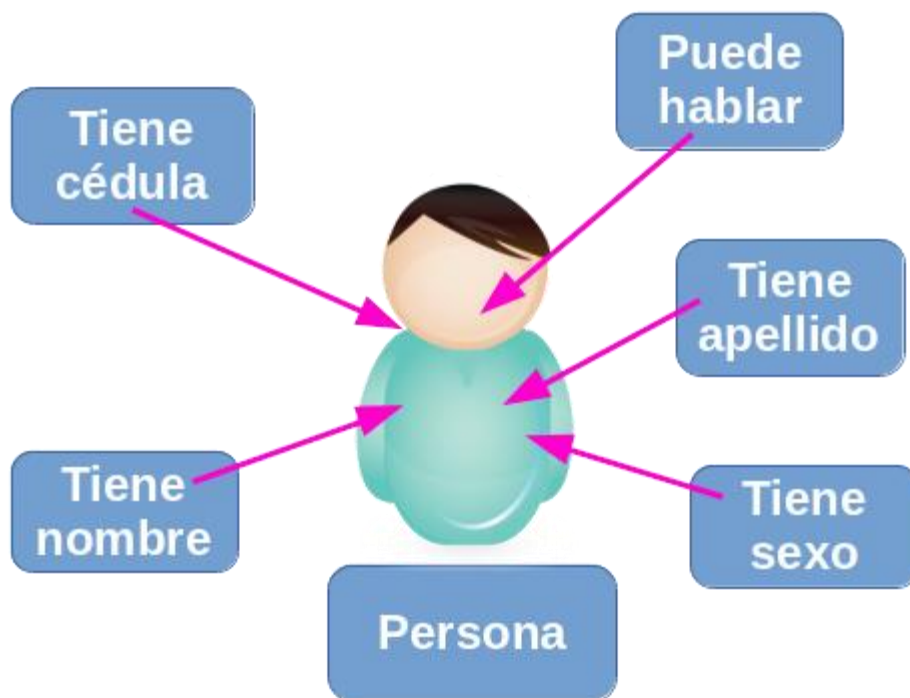


# Herencia

La herencia es una de las premisas y técnicas de la POO la cual permite a los programadores crear una clase general primero y luego más tarde crear clases más especializadas que reutilicen código de la clase general. La herencia también le permite escribir un código más limpio y legible.

## Clase Base

Clase Base o también conocida como Clase abstracta le permite definir una clase que puede heredarse en otras clases los atributos y comportamientos definidos en esta.



Use el diagrama anterior para ilustrar el concepto de la herencia, vea el caso de dos clases que tienen algo en común, ambas son personas, con atributos de datos personales y comportamientos típicos como hablar, comer, caminar, entonces para eso se crea una clase base llamada Persona. A continuación, un ejemplo de la clase Persona con un método interno:

```

class Persona(object):
    """Clase que representa una Persona"""

    def __init__(self, cedula, nombre, apellido, sexo):
        """Constructor de clase Persona"""
        self.cedula = cedula
        self.nombre = nombre
        self.apellido = apellido
        self.sexo = sexo

    def __str__(self):
        """Devuelve una cadena representativa de Persona"""
        return "%s: %s, %s %s, %s." % (
            self.__doc__[25:34], str(self.cedula), self.nombre,
            self.apellido, self.getGenero(self.sexo))

    def hablar(self, mensaje):
        """Mostrar mensaje de saludo de Persona"""
        return mensaje

    def getGenero(self, sexo):
        """Mostrar el genero de la Persona"""
        genero = ('Masculino', 'Femenino')
        if sexo == "M":
            return genero[0]
        elif sexo == "F":
            return genero[1]
        else:
            return "Desconocido"

```