

## Taller 4

### Preguntas de análisis

Antes de la línea 34:

**A.** ¿Cuál es el peso del carro1? Argumente su respuesta.

**El peso del carro1 es 1 tonelada.**

En la línea donde se crea carro1:

**carro1 = Carro("Tracker")**

No se especifica el argumento peso, y como el constructor (`__init__`) tiene un valor por defecto de 1 para el parámetro peso, se le asigna este valor.

**B.** ¿Cuál es el motor del carro3? Argumente su respuesta

**El motor del carro3 es None.**

En la línea donde se crea carro3:

**carro3 = Carro("Picanto", 2, "Kia")**

No se especifica el argumento motor. En el constructor (`__init__`), el parámetro motor tiene como valor por defecto None, por lo que carro3 usará este valor.

**C.** ¿Cuál es la marca del carro0 (En caso de que este no estuviera comentado en la línea 30)? Justifique lo que sucede.

**La marca del carro0 sería "Ford".**

En el constructor, si no se especifica el argumento marca, se utiliza el valor por defecto "Ford". Si la línea `carro0 = Carro()` no estuviera comentada, se ejecutaría sin errores porque todos los argumentos del constructor tienen valores por defecto:

**def \_\_init\_\_(self, referencia, peso=1, marca="Ford", motor=None):**

El único argumento obligatorio es referencia, y como no se pasa este argumento en la línea comentada, el programa arrojaría un error diciendo que falta este parámetro obligatorio.

**D.** ¿Qué imprime la línea 35? Justifique.

La línea 35 imprime:

**Sandero Stepway**

La línea 35 ejecuta el método estático `carro_mas_pesado`, que compara los pesos de los carros en la lista `[carro1, carro2, carro3]`:

**print(Carro.carro\_mas\_pesado([carro1, carro2, carro3]))**

1. carro1 tiene un peso de 1 tonelada.
2. carro2 tiene un peso de 3 toneladas.
3. carro3 tiene un peso de 2 toneladas.

El método recorre la lista y encuentra que carro2 tiene el mayor peso (3 toneladas), y retorna su referencia ("Sander Stepway"). Por eso, la línea 35 imprime este valor.

- Modifique el inicializador de la clase de la siguiente manera:

**Línea original**

```
- def __init__(self, referencia, peso = 1, marca = "Ford", motor = None):
```

**Línea nueva**

```
- def __init__(self, referencia, peso = 4, marca = "Ford", motor = None):
```

E. ¿Qué imprime la línea 35 después del cambio? Argumente su respuesta

Después del cambio en el inicializador, la línea 35:

```
print(Carro.carro_mas_pesado([carro1, carro2, carro3]))
```

imprimirá:

Tracker

Como tiene un nuevo inicializador:

```
def __init__(self, referencia, peso=4, marca="Ford", motor=None):
```

Ahora, si no se especifica un valor para peso, este será 4 toneladas.

Pesos de los carros tras el cambio:

- carro1: No se especificó el peso, por lo que toma el valor por defecto de 4 toneladas.
- carro2: Tiene un peso explícito de 3 toneladas (no cambia).
- carro3: Tiene un peso explícito de 2 toneladas (no cambia).

Método carro\_mas\_pesado: Este método encuentra el carro con el mayor peso comparando los valores devueltos por getPeso().

Ahora, carro1 tiene el mayor peso (4 toneladas), por lo que retorna la referencia "Tracker".

