

# Stock de productos

Un negocio necesita realizar algunos chequeos cotidianos sobre los productos que tiene en stock.

Un producto consta de:

- Nombre
- Marca
- País de elaboración
- Teléfono del proveedor
- Dirección del proveedor

# Stock de productos

1) Implementar en Python el tipo producto con las siguientes operaciones:

- `Producto(self, ...)`: crea un nuevo producto.
- `p.nombre()`: devuelve el nombre (str) del producto *p*.
- `p.marca()`: devuelve la marca (str) del producto *p*.
- `p.pais_de_elaboracion()`: devuelve el país de elaboración (str) del producto *p*.
- `p.telefono_proveedor()`: devuelve el teléfono del proveedor (str) del producto *p*.
- `p.direccion_proveedor()`: devuelve la dirección del proveedor (str) del producto *p*.

# Stock de productos

- Operaciones para comparar por `==` (`__eq__`) y por `<` (`__lt__`). Las comparaciones por igualdad deben realizarse por nombre, por marca y por país de elaboración. Las comparaciones por menor deben realizarse con los mismos campos y en ese orden.
- Una operación para representar como string (`__str__`) a un producto, según el siguiente formato:

<nombre - marca - país - teléfono>

# Stock de productos

2) Escribir un programa en Python que lea los productos listados en el archivo stock.csv e imprima por pantalla los distintos países en donde fueron elaborados.

# Stock de productos

3) Recientemente se ha implementado la iniciativa “Precios módicos” y se quiere verificar cuáles de los productos de la iniciativa se encuentran en stock.

Se pide escribir un programa en Python que lea los productos listados en el archivo `stock.csv` y en el archivo `precios_modicos.csv`, e imprima por pantalla todos los productos del stock que pertenecen a la iniciativa.

Recordar que dos productos se consideran iguales si sus nombres, marcas y países de elaboración son iguales.

4) Calcular el orden de complejidad, en el peor caso, del punto (3) en función de la cantidad de productos en el stock.