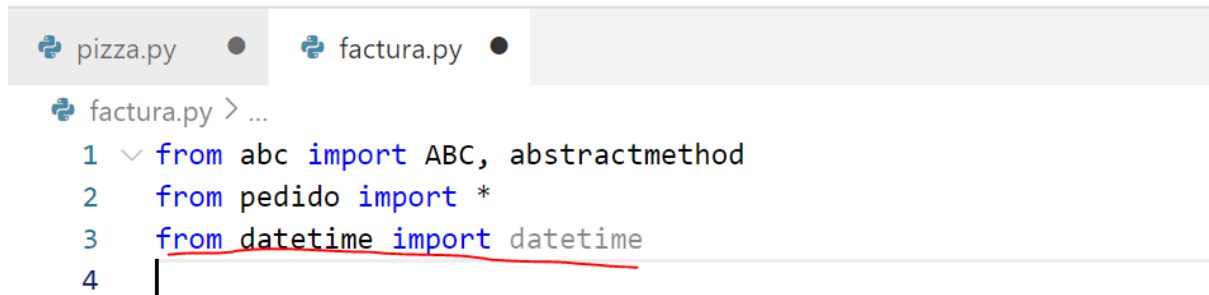


Refactorización

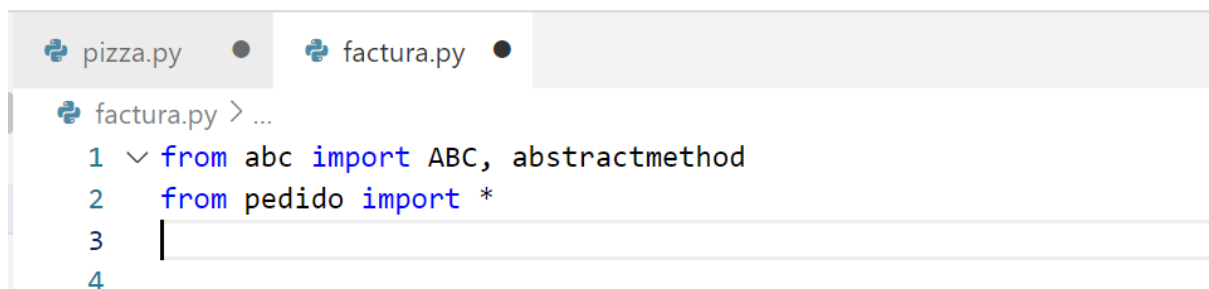
Aquí tenemos ejemplos de código refactorizado:

En la siguiente captura, mostramos el uso de la librería **datetime**, la cual no usamos realmente en el código, como warning nos avisa del no uso de esta librería, por lo cual tenemos que eliminarla de la cabecera.



```
factura.py > ...
1  ∨ from abc import ABC, abstractmethod
2  from pedido import *
3  from datetime import datetime
4  |
```

Como se puede ver en la imagen inferior, hemos eliminado la librería que está sin usar.



```
factura.py > ...
1  ∨ from abc import ABC, abstractmethod
2  from pedido import *
3  |
4  |
```

En las siguientes imágenes se muestra cómo cambiamos el nombre a una variable para que se entienda más fácilmente cuál es su funcionamiento:

```
class pizza(pizza_abstracta):
    #Creamos el constructor
```

```
class pizza_interfaz(ABC):
    #A cada función hay que añadirle '@abstractmethod'
    @abstractmethod
    def agregar_pizza(self, nombre, tamaño, precio):
        pass
    @abstractmethod
    def cambiar_datos_pizza(self, dato_nuevo, tamaño, nombr
        pass
    @abstractmethod
    def eliminar_datos_pizza(self, nombre_pizza, tamaño):
        pass
#Al crear la clase que usaremos especificaremos que heredar
class pizza(pizza_interfaz):
    #Creamos el constructor
```