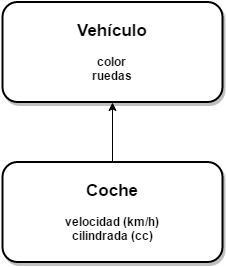
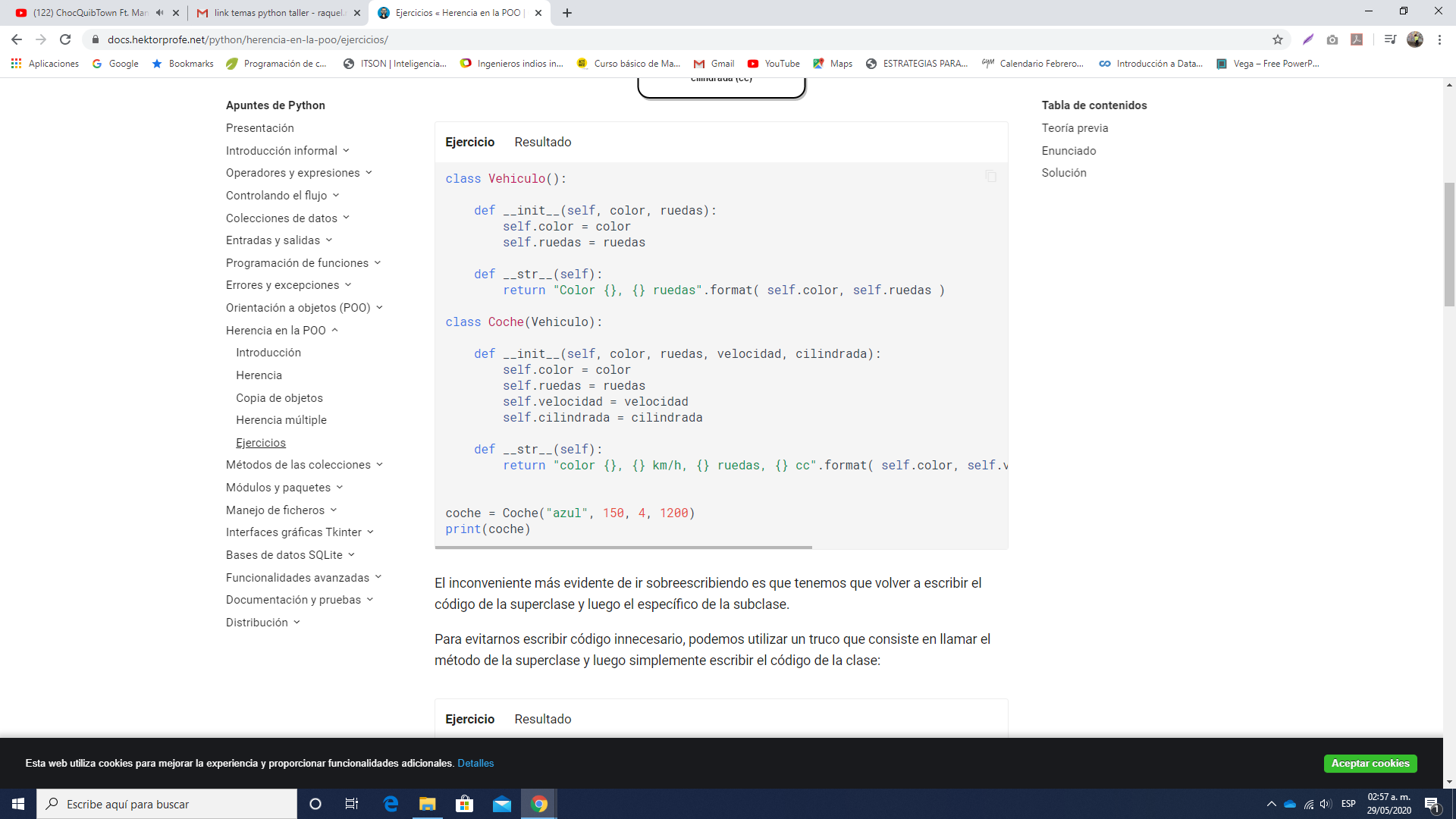
**Practica capítulo 9: Herencia**

## **Teoría previa**

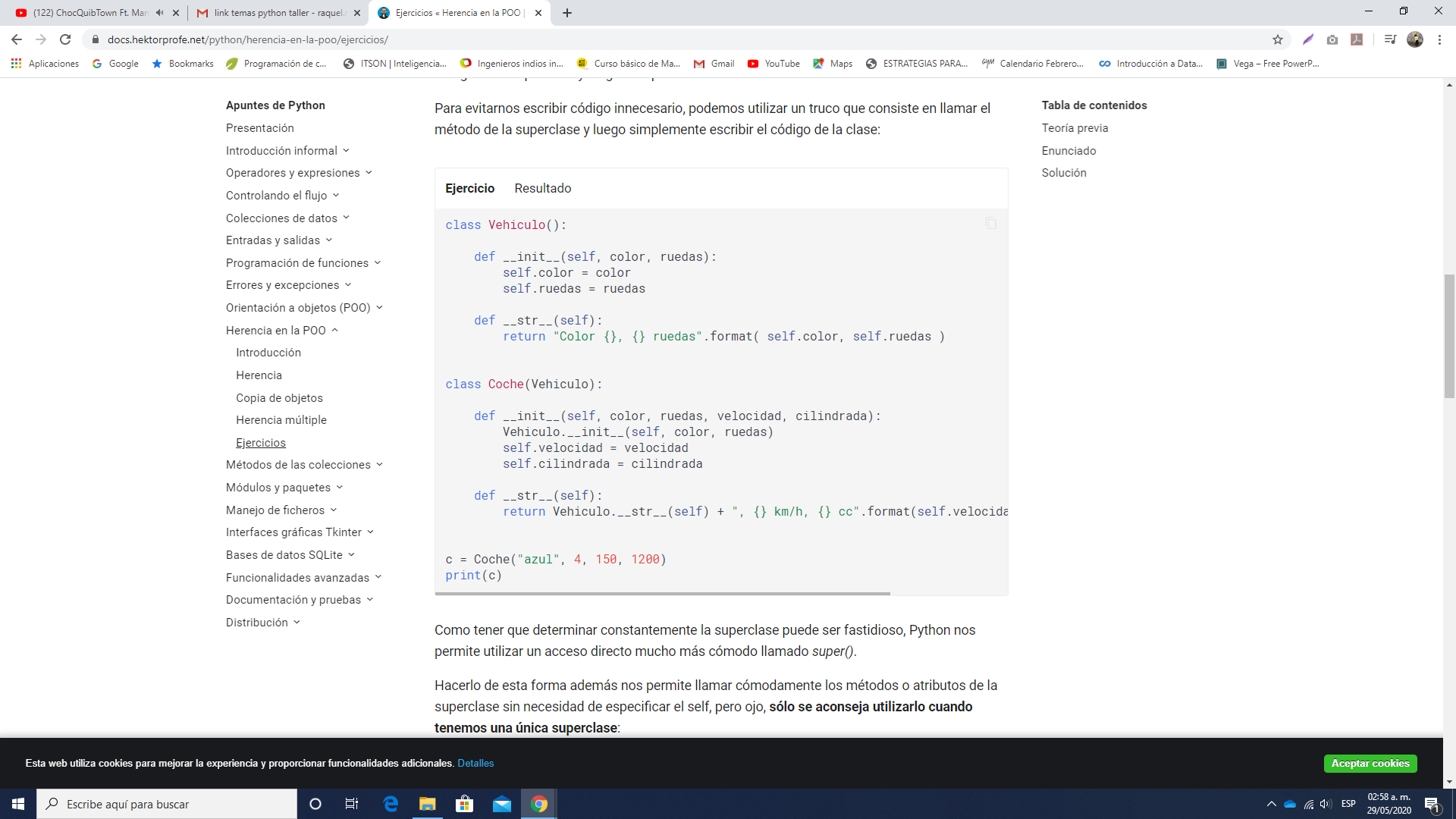
En este ejercicio vas a trabajar el concepto de herencia. Hasta ahora sabemos que una clase heredada puede fácilmente extender algunas funcionalidades, simplemente añadiendo nuevos atributos y métodos, o sobrescribiendo los ya existentes. Como en el siguiente ejemplo:





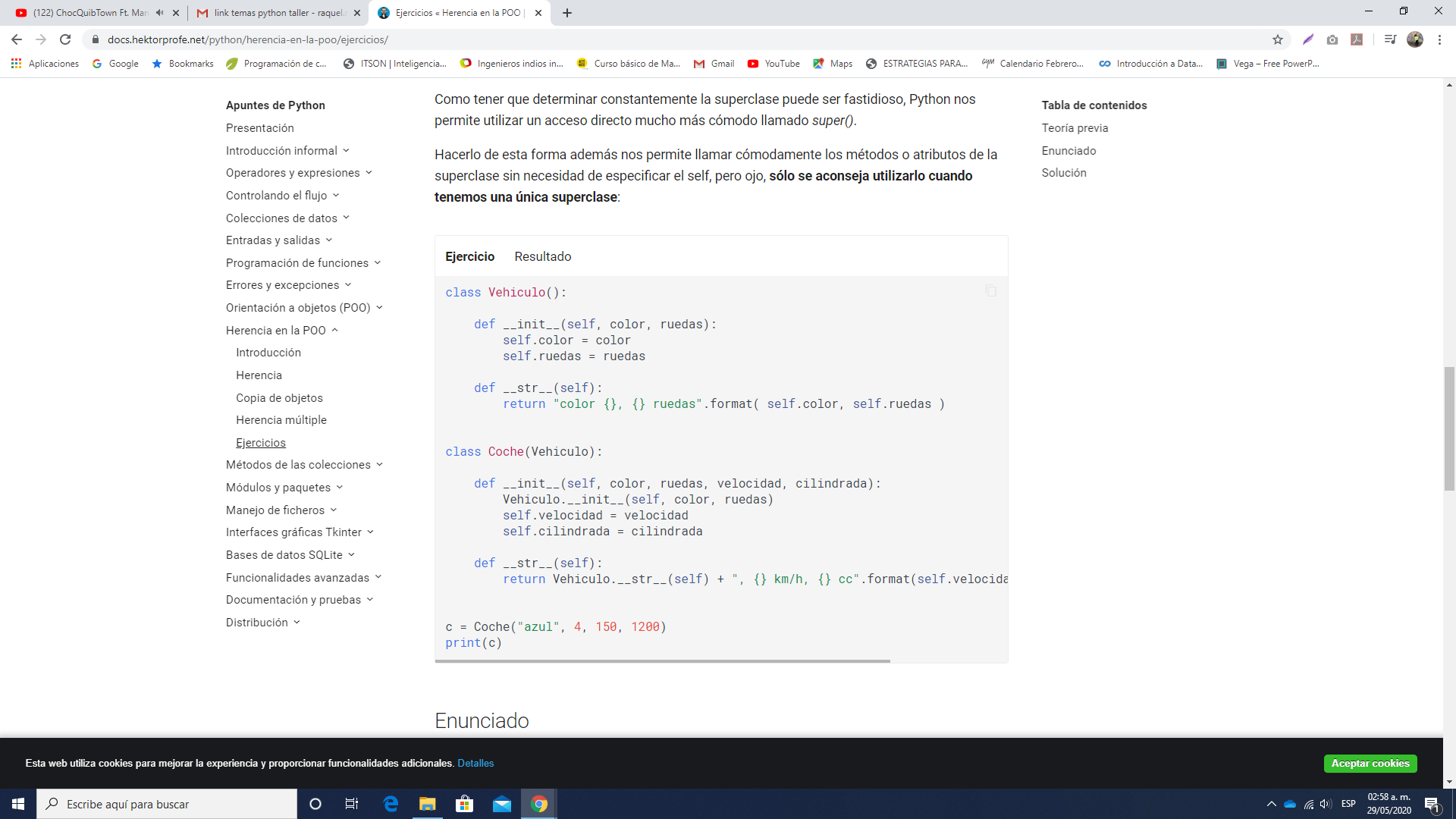
El inconveniente más evidente de ir sobrescribiendo es que tenemos que volver a escribir el código de la superclase y luego el específico de la subclase.

Para evitarnos escribir código innecesario, podemos utilizar un truco que consiste en llamar el método de la superclase y luego simplemente escribir el código de la clase:



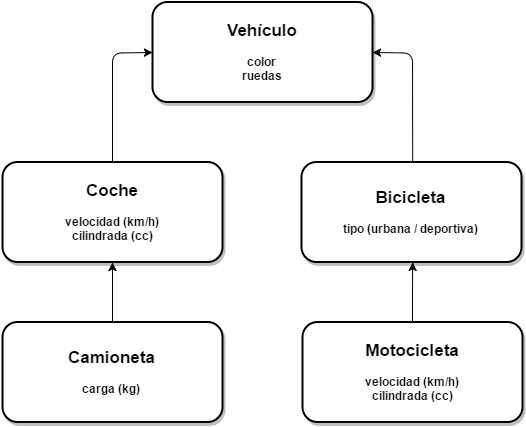
Como tener que determinar constantemente la superclase puede ser fastidioso, Python nos permite utilizar un acceso directo mucho más cómodo llamado *super()*.

Hacerlo de esta forma además nos permite llamar cómodamente los métodos o atributos de la superclase sin necesidad de especificar el self, pero ojo, **sólo se aconseja utilizarlo cuando tenemos una única superclase**:



## **Enunciado**

Utilizando esta nueva técnica extiende la clase Vehículo y realiza la siguiente implementación:



* Crea al menos un objeto de cada subclase y añádelos a una lista llamada vehículos.
* Realiza una función llamada **catalogar()** que reciba la lista de vehículos y los recorra mostrando el nombre de su clase y sus atributos.
* Modifica la función **catalogar()** para que reciba un argumento optativo **ruedas**, haciendo que muestre únicamente los que su número de ruedas concuerde con el valor del argumento. También debe mostrar un mensaje **"Se han encontrado {} vehículos con {} ruedas:"** únicamente si se envía el argumento ruedas. Ponla a prueba con 0, 2 y 4 ruedas como valor.

