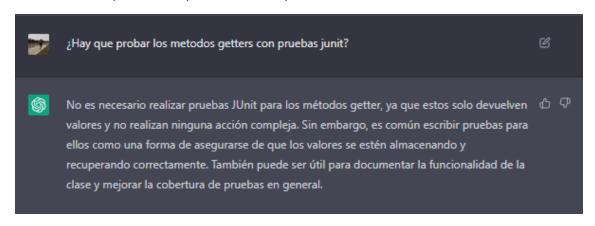
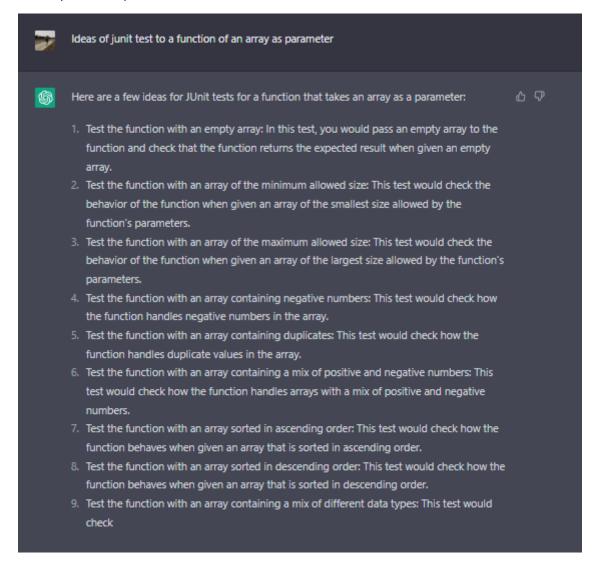
Ya que mi código no llegaba a obtener el 100% de cobertura por culpa de no hacer tests para los métodos getters, le pregunté para saber su opinión al respecto. Al igual que él, opinaba que no eran necesarios pero cierto que es son útiles para la documentación.



A continuación decidí preguntar por si se le ocurrían pruebas junit que yo ignoraba. Alguna como las 2,7,8 las veía poco útiles pero la idea de la 4 (en nuestro caso edades negativas) me pareció buena y decidí implementarla.



Además, una vez realizado los test que creí conveniente, decidí pasarle el código de mi Person.java para ver que pruebas junit me realizaba. Para mi sorpresa generó algunas de las que ya poseía pero ninguna nueva.

```
class PersonTest {
    @Test
         List<Person> persons = null;
         Exception exception = assertThrows(NullListException.class, () -> {
             Person.averageAgePerGender(persons);
         assertEquals("Error, null as persons", exception.getMessage());
    }
    @Test
    void testAverageAgePerGender WithEmp
         List<Person> persons = new ArrayList<>();
         double[] result = Person.averageAgePerGender(persons);
         assertArrayEquals(new double[] {0, 0}, result);
    }
    @Test
         List<Person> persons = new ArrayList<>();
         persons.add(new Person("John", 30, "Male"));
        persons.add(new Person("Jack", 40, "Male"));
persons.add(new Person("Jim", 35, "Male"));
                                on("Jack", 40, "Male"));
         double[] result = Person.averageAgePerGender(persons);
         assertArrayEquals(new double[] {35, 0}, result, 0.0001);
    }
    @Test
         List<Person> persons = new ArrayList<>();
        persons.add(new Person("Jane", 25, "Female"));
persons.add(new Person("Jill", 28, "Female"));
         persons.add(new Person("Joan", 32, "Female"));
             ble[] result = Person.averageAgePerGender(persons);
         assertArrayEquals(new double[] {0, 28.666666}, result, 0.0001);
    }
    @Test
         List<Person> persons = new ArrayList<>();
         persons.add(new Person("John", 30, "Male"));
persons.add(new Person("Jack", 40, "Male"));
         persons.add(new Person("Jack", 40, "Male"));
persons.add(new Person("Jim", 35, "Male"));
         persons.add(new Person("Jane", 25, "Female"));
         persons.add(new Person("Jill", 28, "Female"));
         persons.add(new Person("Joan", 32, "Female"));
         double[] result = Person.averageAgePerGender(persons);
         assertArrayEquals(new double[] {35, 28.666666}, result, 0.0001);
```