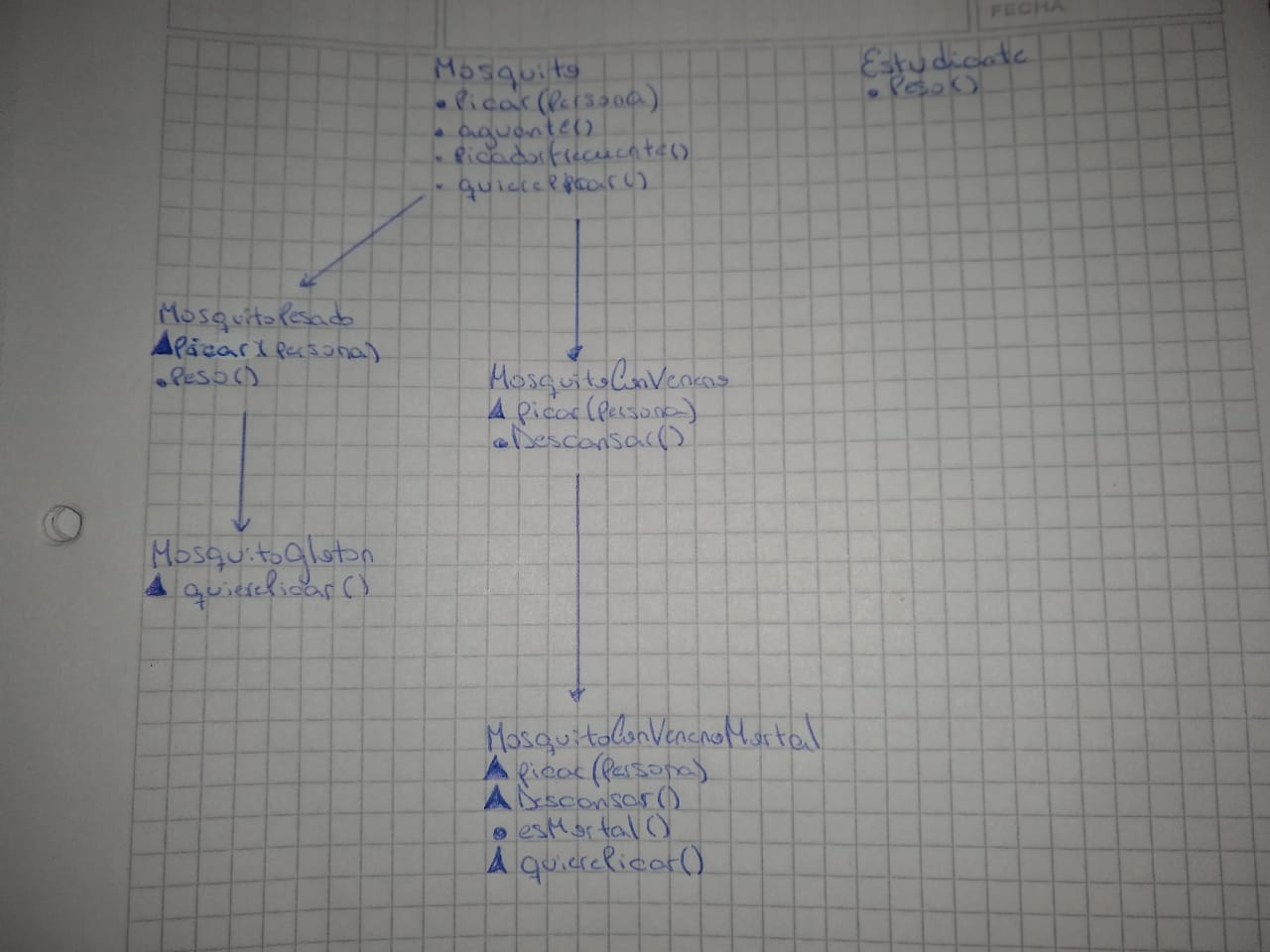
1\_

2\_ mosquito1:

Aguante

CantidadDePicados

Peso

Mosquito2:

Aguante

cantidadDePicados

peso

mosquito3:

aguante

cantidadDePicados

veneno

mosquito4:

aguante

cantidadDePicados

veneno

mortalidad

3\_ en mosquito3 y mosquito4 cambia su valor de aguante restándole el peso de Malena y aumenta en 1 la cantidadDePicados. En mosquito3 aparte de cambiar lo anterior cambia también el veneno se le resta en 1 y el mosquito4 resta su mortalidad en 2.

En mosquito1 y 2 cambia solamente su peso aumentándolo al número que da el peso de Malena dividido por 100.0.

4\_ no quiere picar porque tiene 300 de aguante y no supera los 500 de aguante que necesita para hacerlo. Al tener super() en el método lo hereda de mosquitoConVeneno, como el no modifico el método tiene que preguntarle a mosquito y el método de mosquito dice que da verdadero cuando el aguante es mayor a 500.

5\_ ahora si daría True por que el veneno de mosquitoConVenenoMortal es mayor a 5.

6\_descansar lo entiende mosquito3 y mosquito4.

quierePicar lo entiende mosquito1, mosquito2, mosquito3 y mosquito4.

peso lo entiende mosquito1, mosquito2 y Malena.

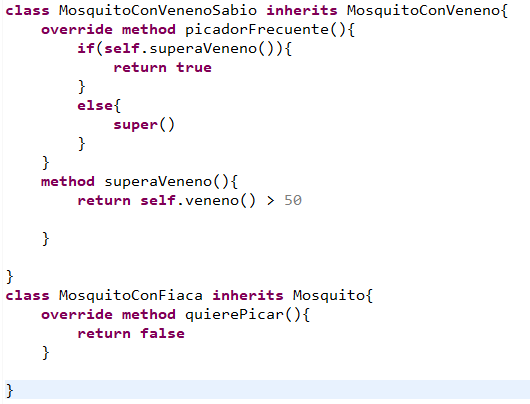
picadorFrecuente lo entiende mosquito1, mosquito2, mosquito3 y mosquito4.

EsMortal lo entiende mosquito4.

7\_ en mosquito1 el aguante es mayor a 500 entonces daría true. En mosquito2 cumple con el aguante pero no cumple pero no cumple con el peso que tiene que ser mayor a 100 entonces da false. El mosquito3 da false por que el aguante es de 150 y no supera los 500. Mosquito4 supera el aguante y también la mortalidad que tiene que ser mayor a 4 entonces da true.

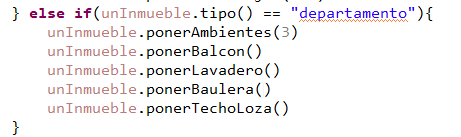
8\_ en mosquito3 devuelve false porque supera la condición que sea la mortalidad mayor a 10 pero no supera en aguante ya que tiene 1800 y necesita que sea mayor a 2000. En mosquito5 devuelve true ya que supera la mortalidad con 50 y su aguante es de 2300 superando los 2000 de condición.

9\_

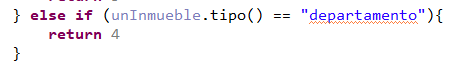


**Inmuebles**

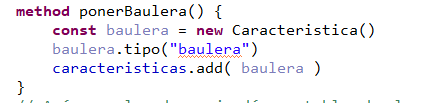
1\_tiene que agregar una condición más al método armar del objeto armado.



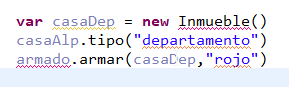
En el método cuantaGentePuedeVivir()de la class inmueble ingresar la condición.



Y poner baúl en la misma clase.



Para crearlo se le hace new a la clase Inmueble() y luego a casaDep se le agrega el tipo “departamento”. Al final se le hace el armado.



2\_mi crítica es que se podrían crear los objetos que sean casa alpina, casa normal, etc. Con un método polimórfico que sea cantDeGente() que devuelva la cantidad de gente que vive en cada casa. Así en el método de la class inmueble cuantaGentePuedeVivir() haga un return con tipo.cantDeGente() y ahí devolvería la cantidad de gente que puede vivir.