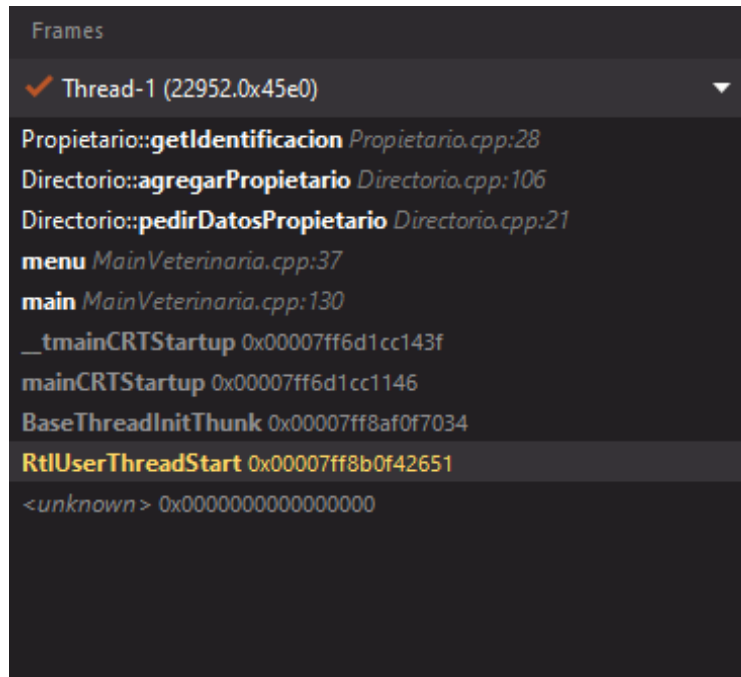


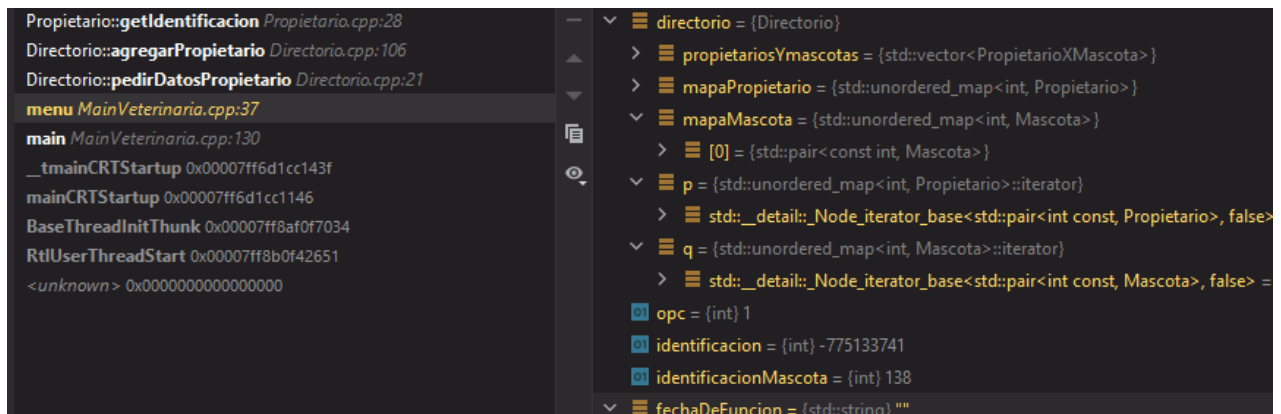
Practica de Debugger en Clion

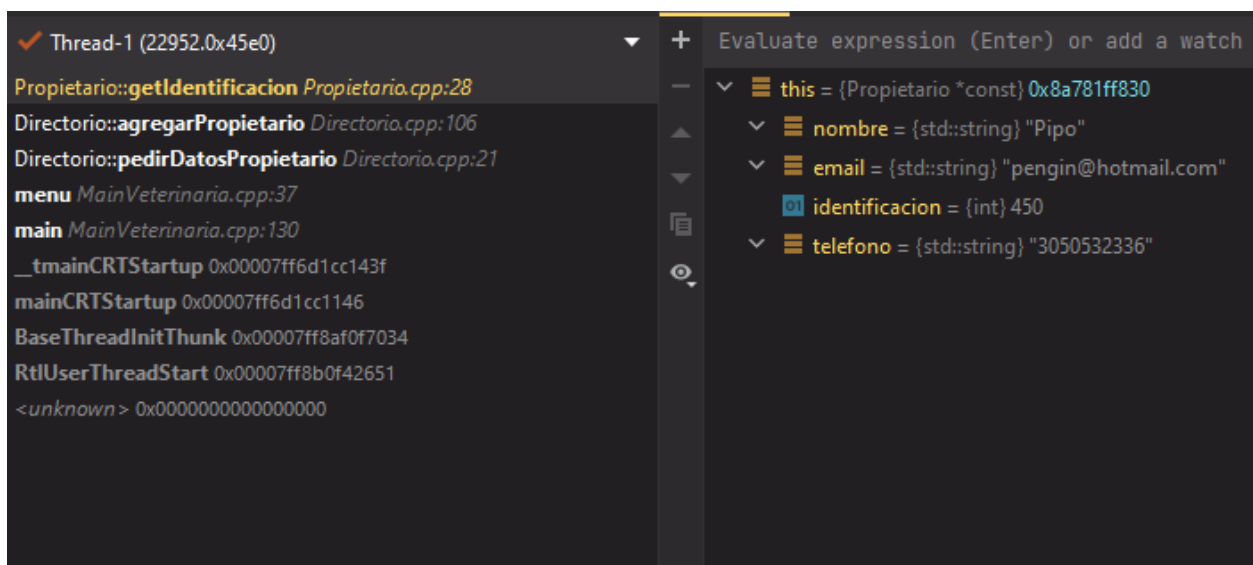
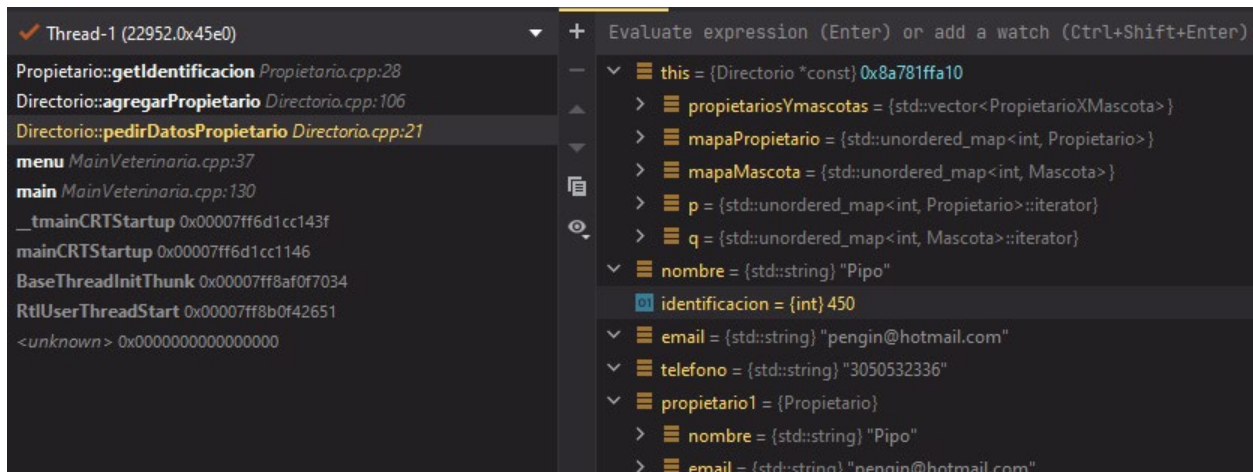
1-)

a)

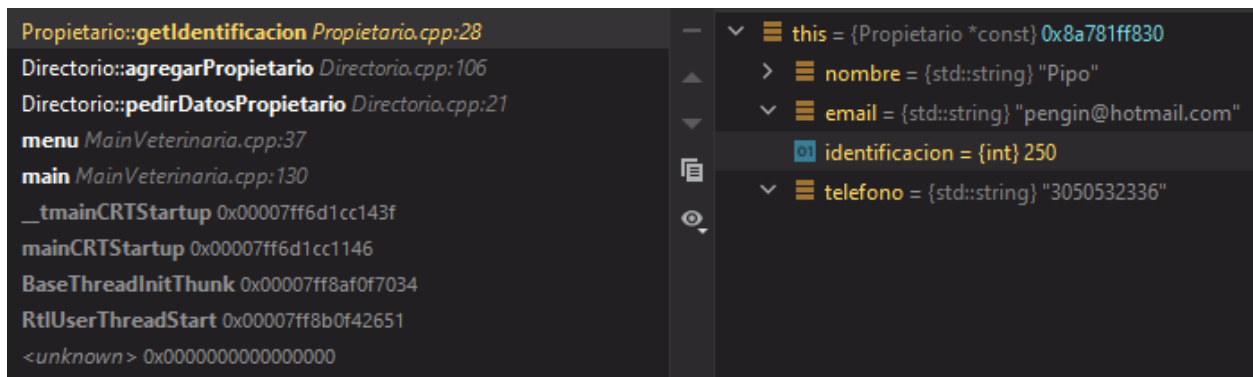


b)



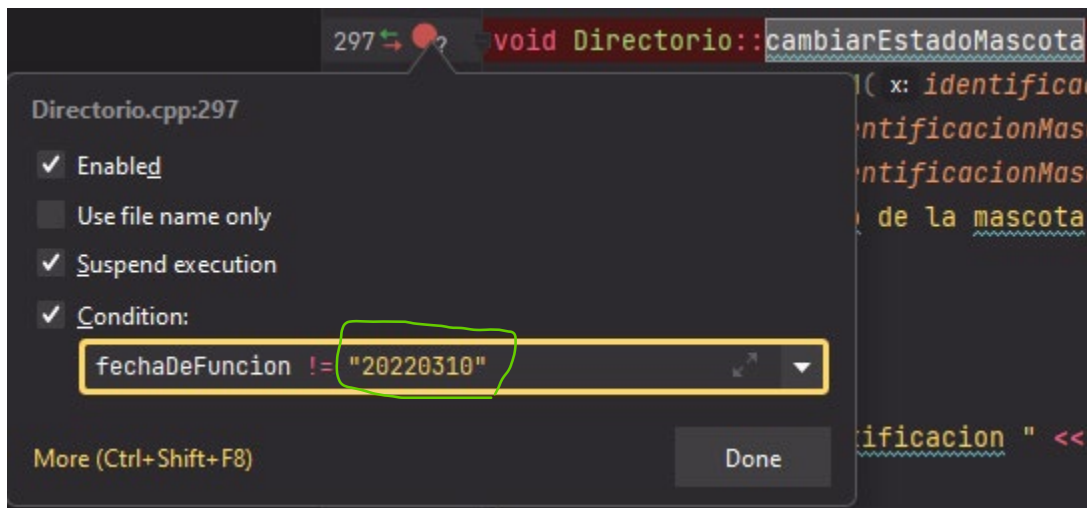


2-) Como en nuestro caso se maneja el getIdentificacion en vez del getNombre, se cambia el valor de identificación de 450 a 250, como aparece en la imagen:

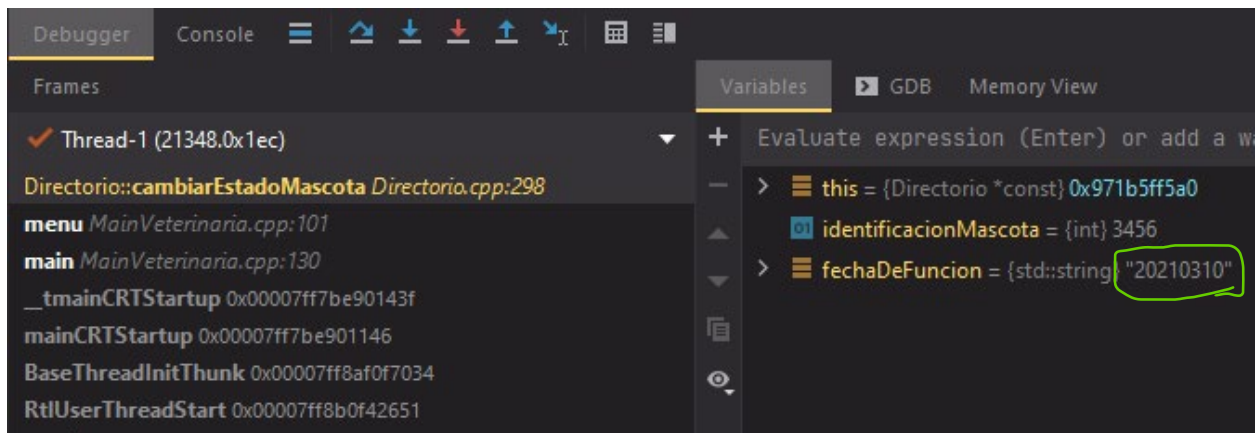


4-)

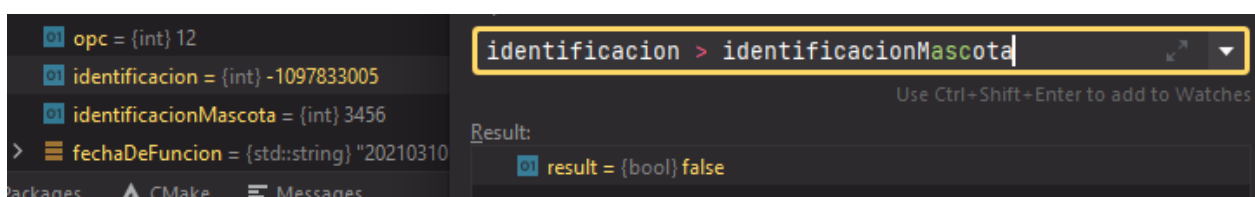
Para este caso decidí probar un condicional en el método *cambiarEstadoMascota*, localizado en *Directorio.cpp*. La condición en sí, es algo exótica y solo es probada con intenciones de curiosidad y de querer saber si estoy haciendo las cosas bien o no.



Después de añadir una mascota y tratar de cambiar su estado, puse la fecha “20210310”, con el cual Clion exitosamente me mandó a las pantallas del breakpoint. Con esto, confirmo que el condicional cumplió su función.

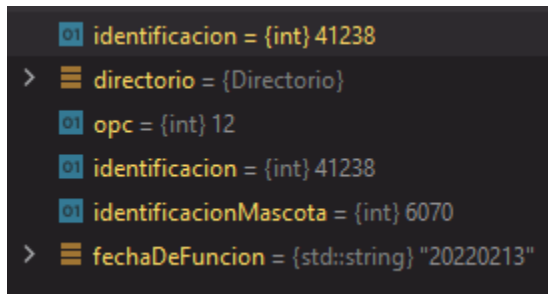


5-)



6-)

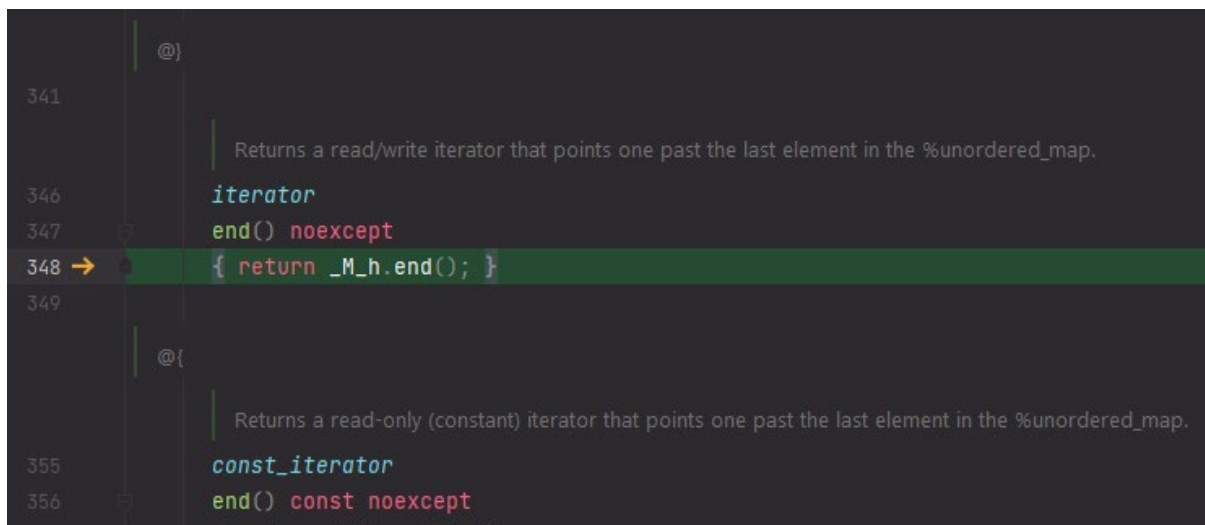
Al darle clic derecho y en la opción “Add to Watches” a una dada variable en el debugger, esta se “ancla” a lo más arriba de la lista, de modo que uno tiene vista prioritaria de los cambios que hay en esa variable a lo largo del procedimiento hecho en el código. Por ejemplo:



```
01 identificacion = {int} 41238
>  directorio = {Directorio}
01 opc = {int} 12
01 identificacion = {int} 41238
01 identificacionMascota = {int} 6070
>  fechaDeFuncion = {std::string} "20220213"
```

7-)

La funcionalidad de “Step Into” es de bastante utilidad al momento que uno necesita comprensión total de que exactamente es lo que se está haciendo en la línea seleccionada. Por ejemplo, utilizando el método al cual se le añadió un breakpoint en el punto 5 y presionando F7 en la misma línea en la que se introdujo, aparece una nueva ventana con la información exacta de que cosas hacen que acciones, todo al pie de la letra.



```
341 | @)
    |
    | Returns a read/write iterator that points one past the last element in the %unordered_map.
346 | iterator
347 | end() noexcept
348 | → { return _M_h.end(); }
349 |
    | @{
    |
    | Returns a read-only (constant) iterator that points one past the last element in the %unordered_map.
355 | const_iterator
356 | end() const noexcept
```

En cambio, la funcionalidad “Step Over” hace lo que su nombre dice. Sin importar que hayan llamadas a funciones, cuando se presiona F8, se salta la línea a la siguiente. Aunque hay que tener en cuenta, que cuando hay breakpoints en medio de las líneas saltadas, el debugger deshabilita la función de saltarse estos.