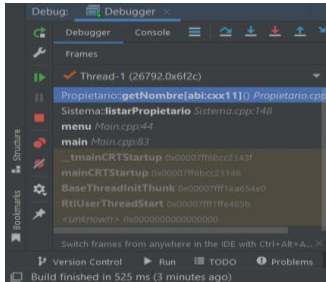


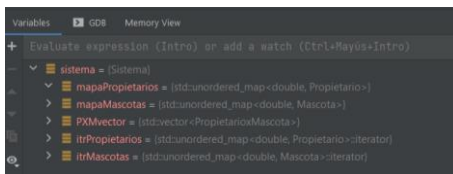
Uso del debugger

1).

a) ¿Cual fue el metodo que inicio toda la ejecucion?

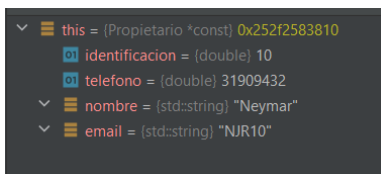


b) Tome una foto que evidencie los valores de las variables de los frames del Main, la clase Sistema/Veterinaria/ o como la hubiera llamado y de la clase Propietario



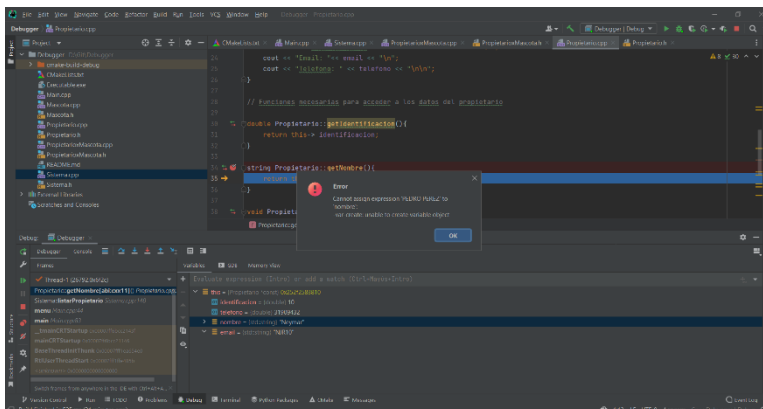
Main

Sistema :: listarPropietario



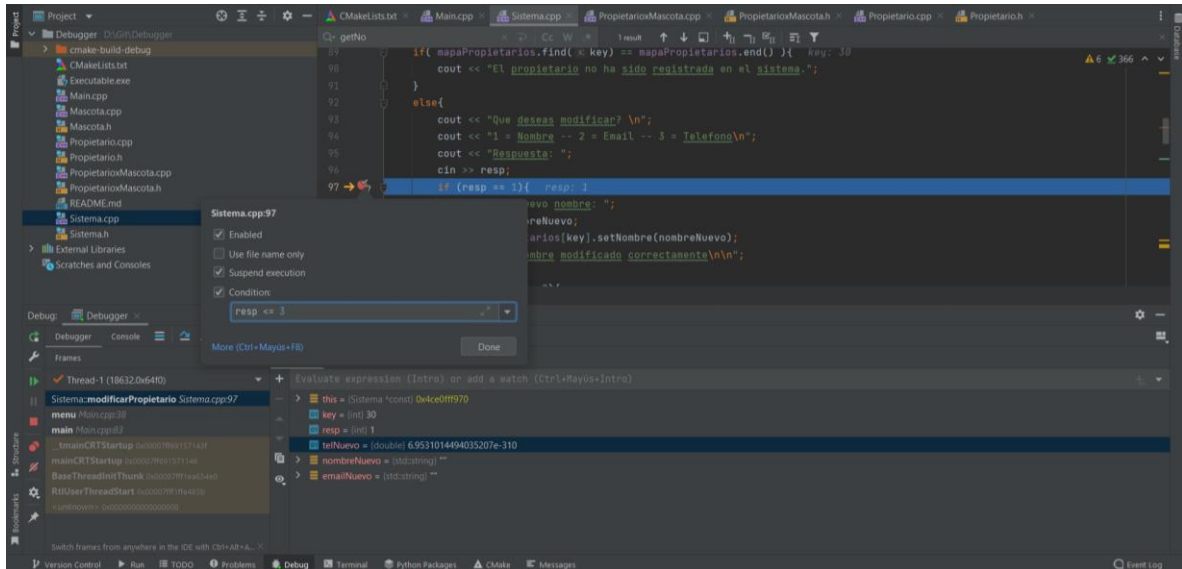
Propietario :: getNombre

2). Set variable values: en el mismo breakpoint, antes de retornar el nombre del propietario cámbielo desde el debug y póngale el nombre de “PEDRO PEREZ”.

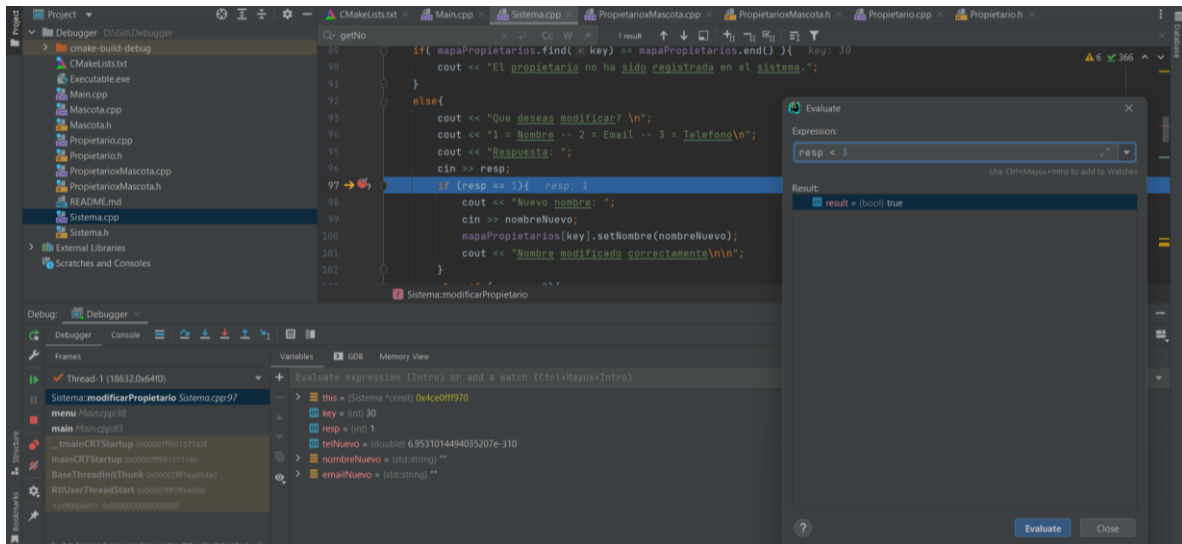


3). BreakPoint condicional: incorpore un breakpoint condicional en el código fuente. Explique en qué funcionalidad lo incorporó. Tome una foto que muestre que el programa se interrumpió exitosamente y muestre el estado de las variables cuando se interrumpió la ejecución.

Lo incorporé para verificar si la respuesta que da el usuario esta dentro de los parámetros deseados.

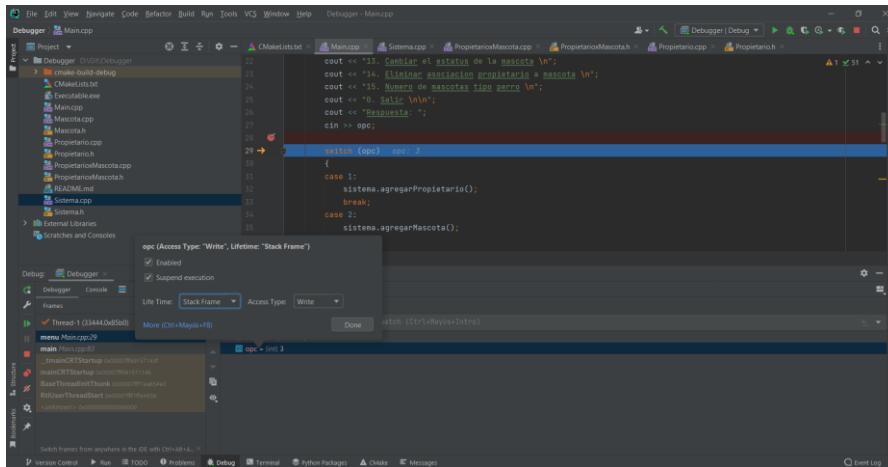


4). Pruebe el evaluador de expresiones con alguna expresión. Tome una foto que muestre que lo probó



5). Agregue un watch para observar alguna de las variables definidas a lo largo del proyecto

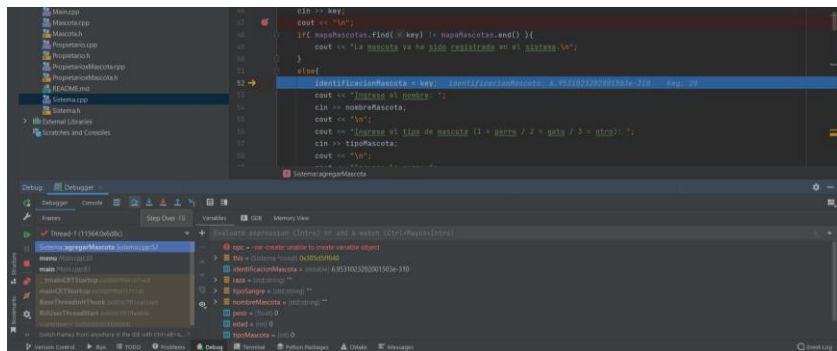
José Manuel García L. (8959600)



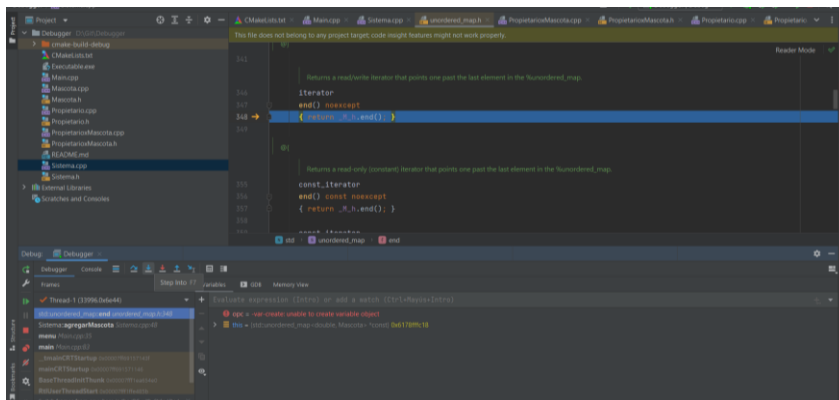
Agregue un watch en el menú para observar la variable “opc” a lo largo del proyecto

6). Que diferencia encontró entre las funcionalidades step over (F8) y la funcionalidad step into (F7). Explique con un ejemplo de las funcionalidades de su proyecto

Step over (F8):



Step into (F7):



Como se ve en las imágenes, el “step over” cuando se encuentra una subfunción en la función (en este caso “end()”), no ingresa a esta subfunción, sino que simplemente la ejecuta y se detiene. Mientras que, por otra parte, el “step into” al encontrarse con esta misma función (“end()”) ingresa en ella haciendo que se abra un archivo nuevo llamado “unordered_map.h” y se detiene en la línea donde se encuentre esa subfunción.