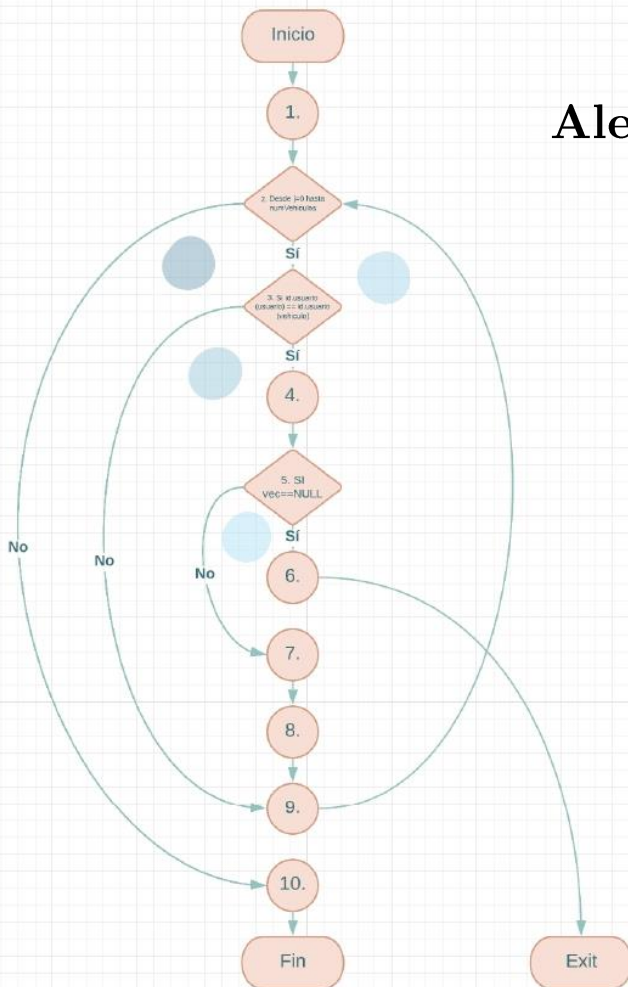


## Alejandro Ruíz del Pino



```

void encontrarVehiculos(Estr_Usuario *usuario, Estr_Vehiculo *vehiculo, int numVehiculos, int **vec, int *x, int i)
{
    int j, h;
    *x=0;

    for(j=0; j<numVehiculos; j++)
    {
        if(strcmp(usuario[i].id_usuario, vehiculo[j].id_usuario)==0) //Si la id del usuario que hemos introducido es igual a la id de
        {
            *vec=(int *)realloc(*vec, ((*x)+1)*sizeof(int)); //signamos un espacio de memoria más, para introducir el vehiculo detectado
            if ((*vec)--NULL)
            {
                printf("Error al asignar memoria.\n");
                exit(1);
            }
            (*vec)[*x]=j; //Se introduce la posición del vehiculo encontrado en la estructura en el vector de enteros dinámico.
            (*x)++; //Vamos aumentando en la cantidad de vehiculos que tiene el usuario, que al mismo tiempo servirá para acceder
        }
    }
}
  
```

- $V_{cg1} = N_a - N_h + 2 = 13 - 10 + 2 = 5$
- $V_{cg} = N_{reg} = 4$
- $V_{cg} = N_{id} + 1 = 3 + 1 = 4$

1. 1, 2, 10

2. 1, 2, 3, 9, 2, 10

3. 1, 2, 3, 4, 5, 6, exit

4. 1, 2, 3, 4, 5, 7, 8, 9, 2, 10

## Pruebas de caja negra

Cabecera: void encontrarVehiculos(Estr\_Usuario \*usuario, Estr\_Vehiculo \*vehiculo, int numVehiculos, int \*\*vec, int \*x, int i);

Precondición: usuario, vehiculo, vec y x cargados e inicializados

Postcondición: Encuentra los vehículos que posee un usuario dado.

Este proceso se completa una vez que se han comparado los dueños de cada vehículo registrado con la ID del usuario que se desee. Si uno de estos coches tiene una ID de vehículo que coincide con la del usuario se tiene en consideración para poder ser nombrado o listado en otra función. Si no encuentra ninguna conexión, el usuario no posee vehículos registrados.

En cuanto a las 4 rutas que se pueden tomar en esta función:

- **1:** Comienza inicializando las variables. Luego entra en el for, el cual repetirá el bucle hasta que j llegue a tener el mismo valor o superior a numVehiculos. En este caso, j si es igual o superior a numVehiculos por lo que salta al final de la función donde finaliza.
- **2:** Volvemos a inicializar las variables, pero entrando dentro del for esta vez. Pasamos por un if que compara si el registro de usuario id.usuario coincide con el registro vehiculo id.usuario. En este caso no son iguales por lo que salta al fin del si, regresando al for inicial y más adelante finalizando el programa pues este for se completa.
- **3:** En este caso, tras inicializar las variables, pasar por el for y llegar al if que compara ambos registros si se da que ambos son iguales, por lo que al entrar dentro de este if y realizar las operaciones que se encontraban dentro llegan a otro if que observa si vec no es nulo. En este caso sí lo es, entrando dentro donde se encuentra un exit(1) que finaliza la ejecución de esta función.
- **4:** En este último caso se repeticen los mismos pasos que en el anterior camino. Sin embargo, esta vez en el if donde comprueba si vec es nulo se comprueba que no lo es, saltando al siguiente nodo hasta llegar al noveno donde vuelve al for del principio, del cual permite salir y finalizar la función.