

Universidad Autónoma de Baja California



Ingeniería en Software y Tecnologías Emergentes.

Práctica 5. Clases de almacenamientos

Materia: Lenguaje C

Maestro: Yulith Altamirano

Alumno: Diego Quiros 372688

Fecha: 09/30/2023

Instrucciones:

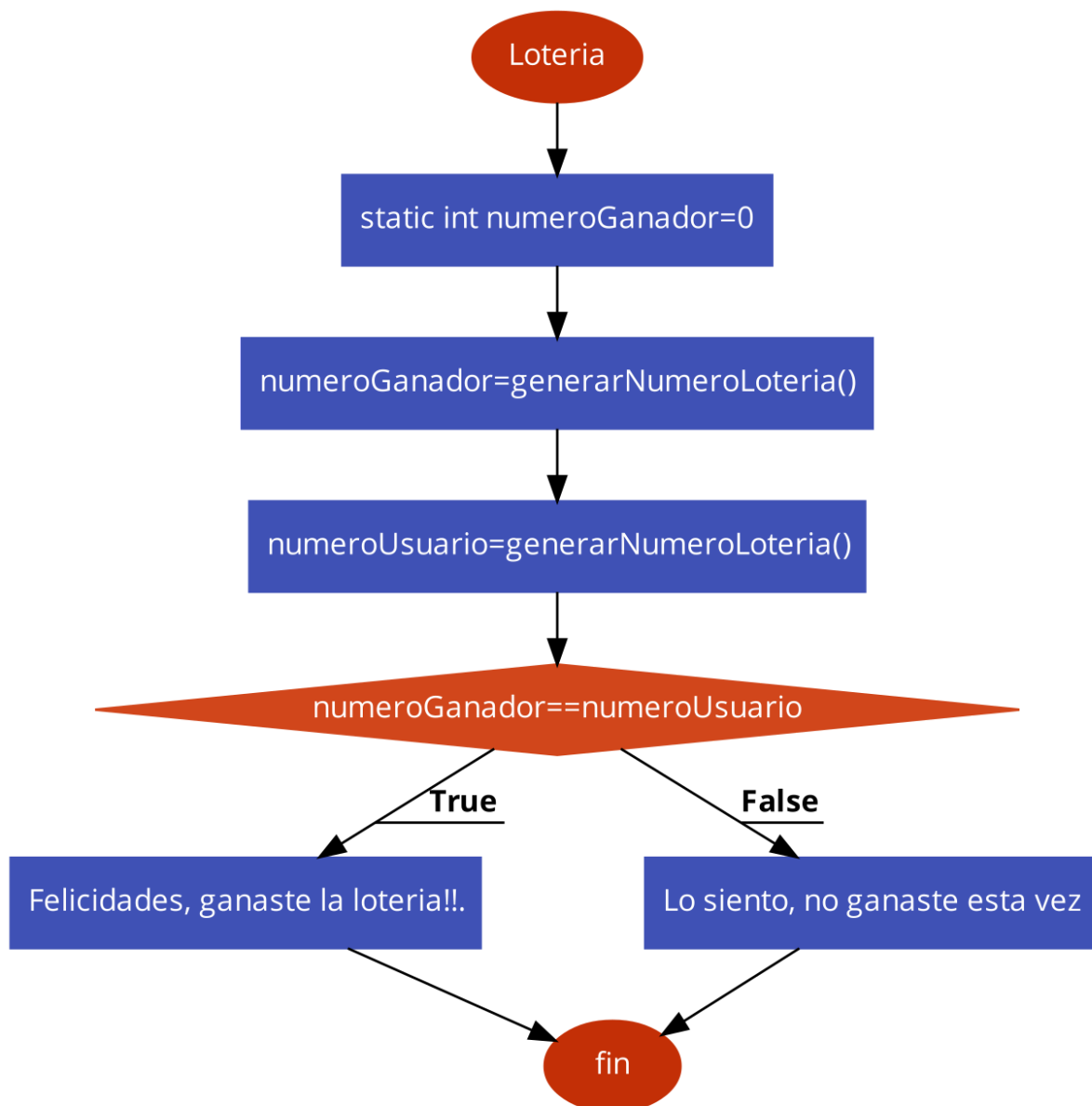
Desarrollen el código en lenguaje C y elaboren el diagrama de flujo correspondiente para los ejercicios. Será suficiente con un archivo .cpp que contenga todos los ejercicios organizados en un menú implementado mediante una estructura switch.

Repositorio:

C/C++

https://github.com/diegovq12/LenguajeC-QVDD_932/tree/main/Practica%205

Diagrama de flujo problema 1:



Problemas ejercicio 1:

El juego de la lotería: Simular un juego de lotería utilizando variables estáticas y automáticas en C.

o Crea una función llamada generarNumeroLoteria que genere un número aleatorio entre 1 y 100 (puedes usar la función rand() y srand() para esto).

```
C/C++
int generarNumeroLoteria(void);
{
    return rand()%100+1;
}
```

o Utiliza una variable estática llamada numeroGanador para almacenar el número ganador. Esta variable debe ser inicializada solo una vez al comienzo del programa.

```
C/C++
static int numeroGanador =0;
```

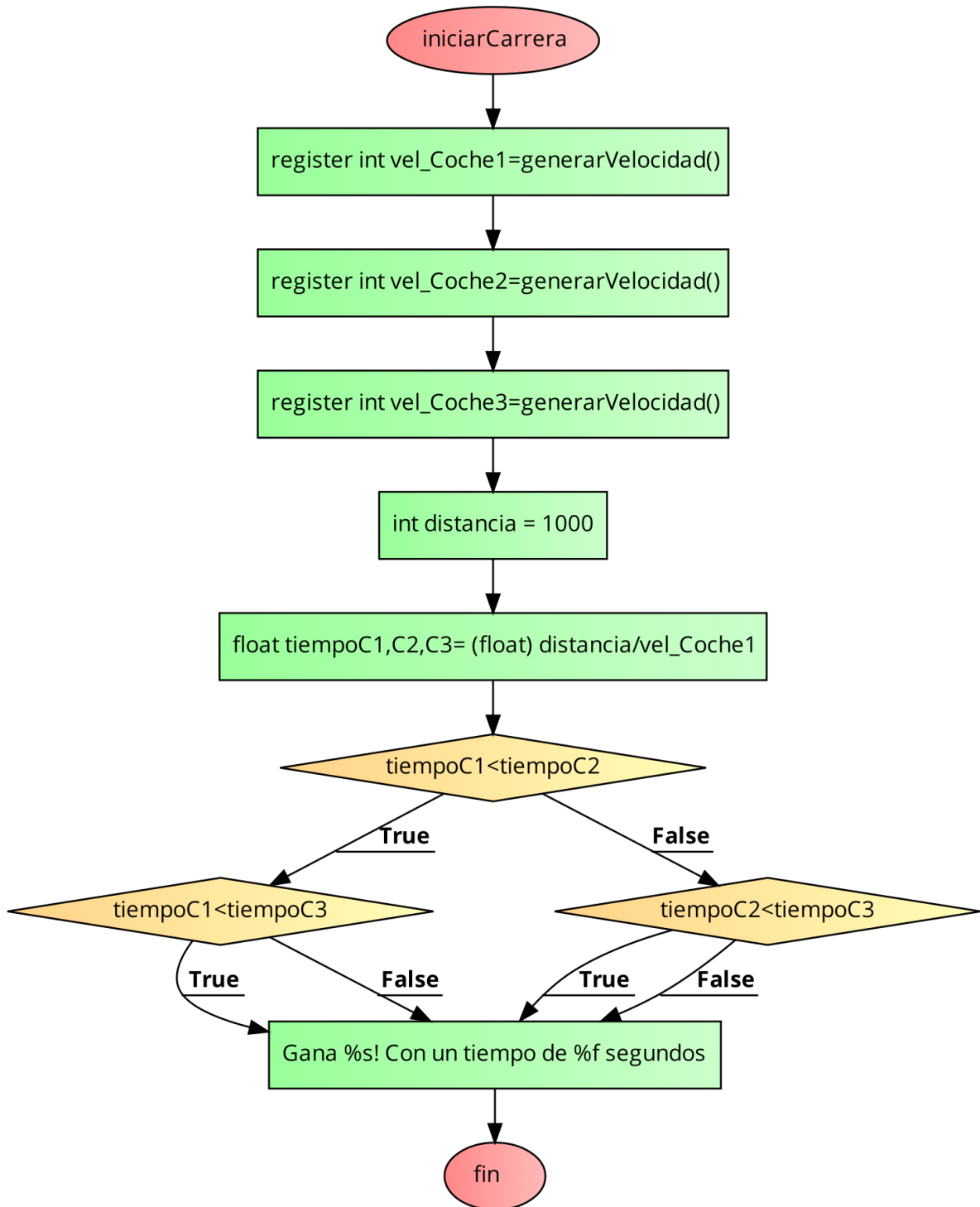
o En el main(), llama a generarNumeroLoteria para obtener el número de lotería de un jugador.

```
C/C++
auto int numeroUsuario=generarNumeroLoteria();
```

o Compara el número del jugador con el numeroGanador. Si coinciden, imprime "¡Felicidades, ganaste la lotería!". De lo contrario, muestra "Lo siento, no ganaste esta vez."

```
C/C++
if (numeroGanador==numeroUsuario)
{
    printf("¡Felicidades, ganaste la loteria!!.");
}
else
{
    printf("Lo siento, no ganaste esta vez.");
}
```

Diagrama de flujo problema 2:



Problemas ejercicio 2:

La Carrera de Coches: Variables de registro (register) en C para simular una carrera de coches.

o Crea una función llamada iniciarCarrera que simule una carrera de coches.

o Declara tres variables de registro (register) para representar a tres coches diferentes.

C/C++

```
//funcion que genera un valor aleatorio entre 100 y 200
int generarVelocidad (void)
{
    return rand() % 101 + 100;
}
register int vel_Coche1=generarVelocidad();
register int vel_Coche2=generarVelocidad();
register int vel_Coche3=generarVelocidad();
```

o Asigna nombres ficticios a cada coche, como "Coche1", "Coche2" y "Coche3".

C/C++

```
char coche1[15] = "Ferrari";
char coche2[15] = "Red Bull";
char coche3[15] = "Honda";
```

o Utiliza las variables de registro para almacenar las velocidades de cada coche. Genera velocidades aleatorias entre 100 y 200 km/h para cada coche.

o Simula la carrera calculando el tiempo que le toma a cada coche recorrer una distancia ficticia (por ejemplo, 1000 metros) a su velocidad respectiva.

o Muestra los resultados de la carrera, incluyendo los nombres de los coches, sus velocidades y el tiempo que tardaron en cruzar la línea de meta.

C/C++

```
int distancia = 1000;
//Calcular el tiempo que les tomara a los coches recorrer la pista
float tiempoC1= (float) distancia/vel_Coche1;
float tiempoC2= (float) distancia/vel_Coche2;
float tiempoC3= (float) distancia/vel_Coche3;

printf("Tiempo %s : %.2f\n",coche1,tiempoC1);
printf("Tiempo %s : %.2f\n",coche2,tiempoC2);
printf("Tiempo %s : %.2f\n",coche3,tiempoC3);
```

o Identifica al coche ganador (el que cruza primero la línea de meta) y muestra su nombre como el ganador de la carrera.

C/C++

```
//Comparar los tiempos para determinar un ganador
//Gana el que haga menos tiempo
if (tiempoC1<tiempoC2)
{
    if (tiempoC1<tiempoC3)
    {
        //ganaC1
        printf("Gana %s! Con un tiempo de %f segundos\n",coche1,tiempoC1);

    }
    else
    {
        //ganaC3
        printf("Gana %s! Con un tiempo de %f segundos\n",coche3,tiempoC3);

    }
}
else
{
    if (tiempoC2<tiempoC3)
    {
        //GanaC2
        printf("Gana %s! Con un tiempo de %f segundos\n",coche2,tiempoC2);
    }
    else
    {
        //ganaC3
        printf("Gana %s! Con un tiempo de %f segundos\n",coche3,tiempoC3);
    }
}
}
```