

A dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from this bar, containing the date.

31-7-2020

# Ruleta

Trabajo Final Taller de Programación I

Several thin, dark blue wavy lines originate from the bottom left corner and curve upwards and to the right.

Integrantes: Guido Cerioni y Luis Andrés Zapata Icart

## **Idea general del algoritmo:**

El mismo despliega una lista con las posibles apuestas a llevarse a cabo, una vez realizadas, se prosigue a consultar que valor de ficha usar y que cantidad de la misma.

Teniendo esta información, se pasa a la función principal (realizarApuestas) de la primera parte, la cual consiste en analizar qué tipo de apuesta es la que va a realizar el jugador y esta devuelve una tabla (array de 2 dimensiones) con la o las apuestas realizadas.

Una vez formado el array, este es consumido por la segunda función principal (giraRuleta), que es la que se encarga de generar un numero random simulando el “giro” de la ruleta, seguido a esto, esta analiza si el numero “ganador” coincide con alguna de las apuestas realizadas. De ser así, genera un aviso al usuario de con que apuestas ha ganado y guarda en variables el dinero que va entregando, de lo contrario, comunica al usuario que perdió la mano y se guarda en otra variable el dinero que ha ganado la mesa.

## **Detalle de las variables y arrays más importantes:**

APUESTASTOTALES[][] → array 2D utilizado para guardar la lista de todas las jugadas de cada ronda.

### **Variables que se utilizan para rellenar APUESTASTOTALES:**

tipoapuesta → var. utilizada para almacenar el tipo de apuesta elegida.

cantfichas → var. utilizada para almacenar la cantidad de fichas a apostar.

valorfichas → var. utilizada para almacenar el valor de la ficha a utilizar.

numeroelegido → var. utilizada para almacenar el numero elegido (solo en caso de apuesta a pleno).

numeroGanador → var utilizada para guardar el numero generado de manera random.

columnaUno[] → array usado para almacenar los valores pertenecientes a la columna 1.

columnaDos[] → array usado para almacenar los valores pertenecientes a la columna 2.

columnaTres[] → array usado para almacenar los valores pertenecientes a la columna 3.

colorRojo[] → array usado para almacenar los valores pertenecientes al color rojo.

colorNegro[] → array usado para almacenar los valores pertenecientes al color negro.

nrosPares[] → array usado para almacenar los valores pertenecientes a los números pares.

nrosImpares[] → array usado para almacenar los valores pertenecientes a los números impares.

historialPago → var. usada para almacenar los valores pertenecientes al historial de pago de la mesa.

historialGanancia → var. usada para almacenar los valores pertenecientes al historial de ganancias de la mesa.

### **Funciones más importantes:**

**int** cantfichasApuesta(): recupera la cantidad de fichas a apostar. No recibe ningún parámetro. Devuelve la cantidad de fichas que el usuario quiere apostar.

**int** valorfichasApuesta(): recupera el valor de las fichas a apostar. No recibe ningún parámetro. Devuelve el valor de las fichas que el usuario quiere apostar.

**void** realizarApuestas (int\*, int\*, int\*, int\*): genera el array con las apuestas efectuadas. Recibe como parámetros los punteros de las 4 variables que se utilizan para generar la tabla de apuestas.

**int** contieneValor (int, int [], int): corrobora que el numero ganador este dentro del array en cuestión. Recibe como parámetros el valor ganador, el array a comparar, y el tamaño de este array. Devuelve 0 si el valor no estaba dentro del array o 1 si lo estaba.

**void** giraRuleta(int [10][4]): genera número random, luego comprueba si hay algún ganador y acumula las ganancias/perdidas de la mesa. Recibe como parámetro la tabla con las apuestas.