



INSTITUTO
POLITÉCNICO
SUPERIOR - UNR

Nombre y Apellido: **PORCELLI, Fabricio**

Comisión: 1

Nombre y Apellido: **PERALTA QUINTEROS, Nicolas Cruz**

Comisión: 2

Analista Universitario en Sistemas

TEMA: **“Trabajo Final: 7 y medio”**

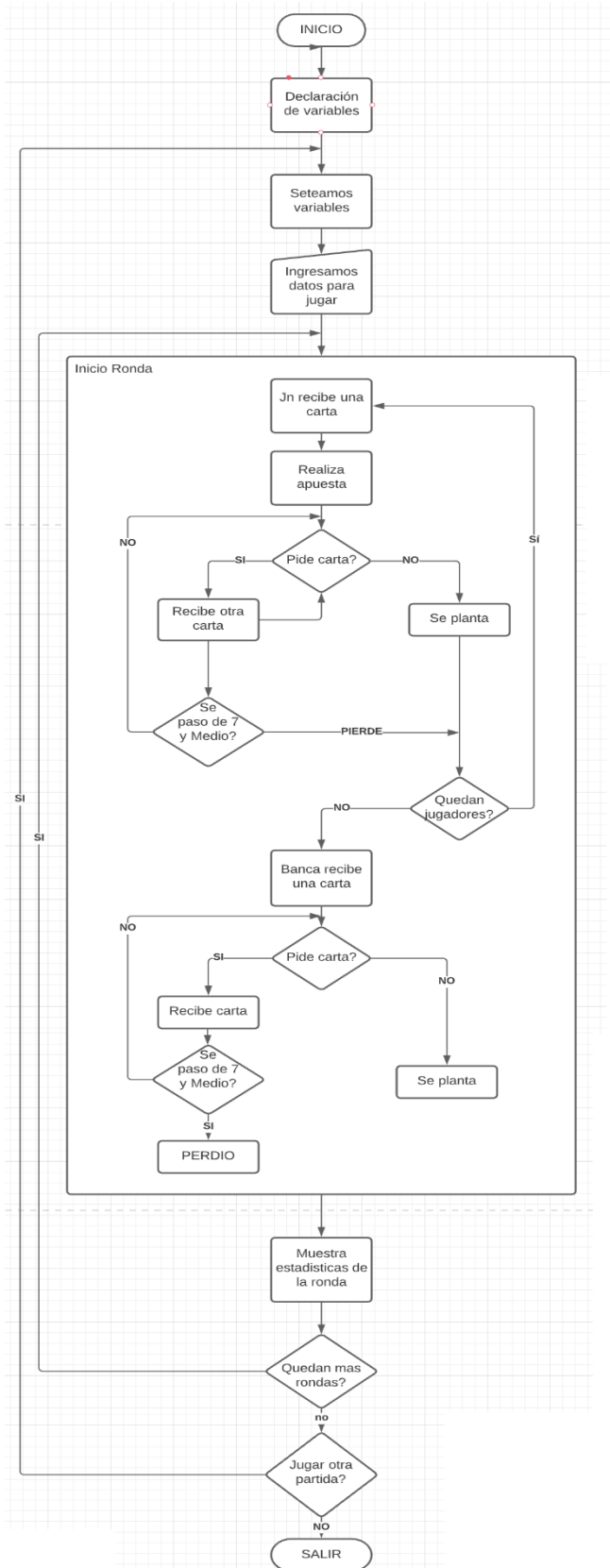
FECHA DE ENTREGA: 30 / 06 / 2021

Consignas

Debe entregarse una documentación digital donde se exprese:

1. Idea general del algoritmo
2. Detalle de las variables y arrays más importantes
3. Comentarios sobre las funciones más importantes

1-



2- Arreglos y Matrices más importantes.

La variable "cartas" que va desde 1 a 40, poniendo el orden a las cartas.

cartasRepartidas [6][28]

	0	1	2	3	4	5	26	27
0								
1								
2								
3								
4								
5								

cantJug + Banca

cartasRepartidas [i][0]: columna nº de los jugadores + banca.

cartasRepartidas [i][1] y **cartasRepartidas [i][1]+2:** columnas que poseen los valores de las cargas del 1-10

cartasRepartidas [i][2] y **cartasRepartidas [i][2]+2:** columnas que poseen los valores de los palos del 1 al 4

apuestas [6][6]

	0	1	2	3	4	5
0						
1						
2						
3						
4						
5						

cantJug + Banca

apuestas [i][0]: columna nº de los jugadores + banca.

apuestas [i][1]: columna pozo del jugador.

apuestas [i][2]: columna apuesta realizada por ronda.

apuestas [i][3]: col. apuesta máx. realizada durante el juego.

apuestas [i][4]: col. ronda en la que realizo la apuesta max.

apuestas [i][5]: columna ganancia acumulada.

contValores [10][3]

	0	1	2
0			
1			
2			
3			
4			
5			
6			
7			
8			
9			

contValores [i][0]: columna que muestra los valores de las cartas del 1-10

contValores [i][1]: columna que muestra la cant. de veces que salio una carta

contValores [i][2]: columna auxiliar.

contGanada [6][2]

	0	1
0		
1		
2		
3		
4		
5		

contGanada [i][0]: columna nº de los jugadores + banca.

contGanada [i][1]: columna cantidad de rondas ganadas.

sumaCartas [6][5]

	0	1	2	3	4
0					
1					
2					
3					
4					
5					

sumaCartas [i][0]: columna nº de los jugadores + banca.

sumaCartas [i][1]: col. sumatoria valor cartas por ronda.

mazo [4][10]

	0	1	2	3	4	5	6	7	8	9	valor
0											
1											
2											
3											
palo											

Matriz donde se guarda el orden de las cartas barajadas

3- Explicación de las funciones.

Dentro del archivo principal main.c tenemos 3 funciones.

- 1) void matrizCartas0 ();
- 2) void verCartas(int,int);
- 3) void linea(int);

1- **matrizCartas0()**

Inicializa la matriz de "cartasRepartidas()"

2- **void verCartas(int,int)**

Es la encargada de imprimir el valor y el palo de la repartida.

Se almacenan en valor1 y palo1 los valores de la carta (por ejemplo el 7 de oro, valor1 = 7 y palo1 = 1, ya que es el primer palo).

Luego utilizando dos const char * -> puntero a constante cadenas de caracteres, con esos dos valores imprime los equivalentes en valor* y palo* . Siendo ambos:

```
const char *palo[4] = {"Oro", "Copa", "Espada", "Basto"};
```

```
const char *valor[10] = {"Uno", "Dos", "Tres", "Cuatro", "Cinco", "Seis", "Siete", "Sota", "Caballo", "Rey"};
```

3- **void línea(int)**

Una función a la que le paso un numero entero y me imprime esa cantidad entera de "\n".

Aparte del main.c tenemos un archivo con funciones llamado funciones_cartas.cpp (cpp porque utiliza una librería de c++).

En el archivo tenemos 4 funciones. Las 3 principales ordenadas según su ejecución serian:

- 1) void inicioCartas()
- 2) void barajar()
- 3) int mazoCartas()

1- **inicioCartas()**

Inicializa la matriz mazo[4][10] con 0 en todas las direcciones.

2- **barajar()**

Elije las cartas del 1 al 40 en orden y les asigna una fila y columna **aleatoria** a cada carta. Para obtener los valores aleatorios utilizamos la función `time_t` de la librería `time.h` de c++.

`time_t tiempo;` // incluida en `time.h` , `time_t` un entero (generalmente de 32 o 64 bits), que representa el número de segundos desde el inicio del tiempo Unix.

`srand(time(&tiempo));` // siembra el generador de números aleatorios usando de semilla el valor devuelto por `time_t`

`fila = rand() % 4;` // selecciona al azar de entre 0 a 3 y se le asigna a la fila
`columna = rand() % 10;` // selecciona al azar de entre 0 a 9 y se le asigna a la columna

3- **mazoCartas()**

Con las cartas mezcladas en la matriz “mazo” las puedo imprimir en el orden de la variable “carta”. Así que en el orden de carta (1 al 40), a medida que la función es llamada desde el main, me devuelve el valor de la fila y columna de la “carta”.

La 4ta es: **void reinicio_cartas()** que no tiene otra utilidad mas que reiniciar la variable “cartas” a 1.