Instituto Nacional de Aprendizaje (INA) Sede Occidente

Programador(a) de Aplicaciones Informáticas (Implementación de aplicaciones informáticas con programación estructurada)

Segunda Parte del Proyecto (Póker)

Nelson Jimenez Jimenez

Redwin Valverde Castro

La Esperanza, Piedades Norte, San Ramón

**Céd:** 2-0789-0853

**Entrega:** 04/05/2021

# Tabla de Contenidos

Contenido

[Tabla de Contenidos 1](#_Toc70958242)

[Introducción 2](#_Toc70958243)

[Objetivos 3](#_Toc70958244)

[Problema Empresarial 4](#_Toc70958245)

[Análisis del Problema 5](#_Toc70958246)

[Datos de entrada: 5](#_Toc70958247)

[Datos de salida: 5](#_Toc70958248)

[Suposiciones u observaciones: 5](#_Toc70958249)

[Describir la estrategia de solución: 6](#_Toc70958250)

[Determinación de reglas de negocio: 8](#_Toc70958251)

# Introducción

Este trabajo escrito es perteneciente a la segunda etapa del proyecto del módulo programación estructurada, contiene el juego en C++ Póker, la intención del desarrollo de este es, aparte de que es un proyecto a elaborar, resolver cada uno de los problemas presentados por el profesor Nelson y el uso de una buena programación estructurada, además, contiene los problemas empresariales y un análisis detallado de problema a resolver junto con sus datos de entrada, datos de salida, suposiciones u observaciones, una descripción de la estrategia de solución y una determinación de las reglas del negocio.

Resolver este problema conlleva un esfuerzo grande y una dedicación de desarrollo corta, tomando en cuenta las habilidades aprendidas y empleadas en el corto tiempo de aprendizaje elaborado en el módulo actual de programación estructurada que son parte del desarrollo de Software; todo el tiempo invertido será completamente al desarrollo del juego Póker y lograr una estabilidad y conformidad en este.

# Objetivos

Desarrollar, determinar, emplear y resolver posibles respuestas o soluciones para el desarrollo de los problemas empresariales dados por la misma empresa. Analizar y resolver los problemas obtenidos mediante el punto de vista del software para el desarrollo del juego en cuestión. Explicar y determinar correctamente los datos de entrada, los datos de salida, las suposiciones u observaciones, la estrategia de solución y una buena determinación de reglas del negocio como parte de un desarrollo y análisis previo de un proceso adecuado de solución. Dar una mejor adaptación para el usuario mediante los datos de entrada y de salida. Determinar unas adecuadas limitaciones, excepciones y suposiciones para un desarrollo más adecuado, junto con una descripción de estrategia y las reglas del negocio.

# Problema Empresarial

Mediante el uso de C++, se realizará un juego de Póker, en el cuál, se implementarán distintos problemas y alternativas al juego original, en este caso se tomarán unas reglas las cuales anteriormente dichas son: serán de 2 a 4 jugadores, se repartirán dos cartas a cada uno, se pondrán 5 cartas a la mesa y serán comunes para todos los jugadores, una vez entregadas las cartas determinar utilizando las reglas de póker cual es la mejor jugada de cada jugador y quien es el ganador.

Solo se deben asignar cartas válidas (baraja francesa de 52 cartas), los resultados de cada partida deberán guardarse en un archivo, mismo que podrá consultarse cuando el usuario lo desee, no puede haber empates. Todo lo anterior escrito será tratado con dedicación para resolver cada uno de los problemas aparentes, la solución no debe tomar en cuenta las apuestas, a no ser que lo quiera incluir como un extra.

# Análisis del Problema

## Datos de entrada:

* Cantidad de jugadores.
* Nombres de los 4 jugadores si es el caso.
* Si los jugadores quieren jugar otra partida.
* Si quieren ver el historial.

## Datos de salida:

* Cantidad de jugadores.
* Ingresar el nombre del jugador en cuestión.
* Nombre del juego.
* La partida actual.
* Cartas en la mano de cada jugador.
* Cartas en la mesa.
* La jugada de cada jugador.
* Se muestra el ganador.
* Ingresar S si los jugadores desean jugar otra partida.
* Ingresar H si los jugadores desean ver el historial.
* El historial si es el caso.

## Suposiciones u observaciones:

* Poca cantidad de jugadores.
* Se logra hacer un desempate dependiendo de la carta después de cada jugada (ej. Póker: la carta siguiente de las 4 cartas iguales).
* El titulo empleado en el historial se realizará solo una vez, ya que se sobre escribe si se mantiene activo.
* Se crearán dos archivos “.h” para mejor ordenamiento de las funciones requeridas.
* Se creará una función adicional para mostrar las cartas de mayor e iguales a menores para mejor ordenamiento para el/la usuario(a).
* Se reinicia cada matriz para su comprobación de mano en la función.
* La función que determina la mano de cada jugador será mayormente completada por ciclos para recorrer dichas matrices.

## 

## Describir la estrategia de solución:

* Incluir las bibliotecas necesarias.
* Llamar cada uno de los .h necesarios en el programa principal.
* Se solicita la cantidad de jugadores, deben ser de 2 a 4.
* Se crea un vector con la biblioteca “vector” y tendrá de tamaño la cantidad de jugadores.
* Se pide el ingreso de cada uno de los jugadores, serán ingresados en el vector por medio de un ciclo For, si no se ingresa un nombre, se usará un nombre predeterminado. Ej. Jugador 2.
* Se creará un ciclo “Do-While” que tendrá dentro el llamado de una función que será la que contendrá el juego principal.
* Dicha función llevará consigo el vector, la cantidad entera de jugadores y la variable de la ronda actual.
* Se borrará siempre la pantalla y el buffer de entrada.
* Dicho ciclo anterior se repetirá si se ingresa una S, también se pedirá si los jugadores quieren ver el historial de ganes ingresando la tecla H.
* En la función anteriormente mencionada, comenzará con la creación de 6 matrices booleanas de 4x13 espacios, una para el mazo completo, cuatro para cada jugador y una para la mesa.
* Se creará una función para llenar adecuadamente cada una de estas funciones, los parámetros de dicha función son: la matriz a llenar y un dato booleano.
* La matriz del mazo completo contendrá todos sus espacios en “true”, lo que quiere decir que tendrá cada una de las cartas.
* Al contrario del punto anterior, las matrices restantes serán rellenas con datos booleanos “false”, lo que quiere decir que no contienen datos.
* Luego se creará una función para dar cartas sin repetir a las matrices de cada jugador y a la mesa, dicha función tendrá como parámetros la matriz del mazo y una de las demás matrices, generará dos números aleatorios, el primero que determina la fila a escoger de la matriz que será del 0 al 3 y el otro que determina la columna, será del 0 al 12, si la posición generada ya está ocupada, se abrirá un “While” que se va a repetir si los campos ya están ocupados y generará otros dos números aleatorios hasta que se escoja una posición sin ocupar.
* La matriz de la mesa será llevada 5 veces a la función nombrada en el punto anterior y contendrá cartas sin repetir que serán usadas más adelante.
* Luego se llenarán las manos del jugador 1 y jugador 2, repitiendo el proceso anterior pero solo dos veces.
* En el caso de que la cantidad de jugadores sea 3 o 4, habrá un “if” que llene las manos de los jugadores en cuestión dependiendo de la cantidad de jugadores.
* Se mostrará en pantalla el nombre del juego, la partida actual, las 2 cartas de cada jugador y las 5 cartas de la mesa.
* Para mostrar dichas cartas, se creará un función que tendrá como parámetros la matriz del jugador en booleano y un dato booleano en “false” que en este caso mostrará las cartas ya existentes en la mano de cada jugador, dentro de dicha función, se crearán 4 variables “char” que cada una tendrá diferentes caracteres, corazón una “c”, diamante una “d”, trébol una “t” y espada una “s”, esta última no será una “e” por su parecido con la “c”; se crean dos ciclos “for“ para recorrer matrices, el primero recorrerá **i** y va de 0 a 4, el segundo **j** irá de 0 a 12, si la posición **j** actual está entre 0 y 8, se mostrará en pantalla su valor +2 para dar un valor correcto, si **j** es 9, será “J”, si **j** es 10, será “Q”, si **j** es 11, será “K”, si **j** es 12, será “A”, en el caso de **i,** si este se encuentra en la fila 0 mostrará el carácter del corazón, si este se encuentra en la fila 1 mostrará el carácter del diamante, si este se encuentra en la fila 2 mostrará el carácter del trébol, si este se encuentra en la fila 3 mostrará el carácter de la espada.
* Lo mismo con las cartas de la mesa.
* Luego se muestra en pantalla las jugadas.
* Se crean 4 nuevas jugadas para las nuevas manos de cada jugador.
* Se llenan con espacios vacíos o booleanos “false”.
* Luego se crean nuevas variables enteras para las jugadas obtenidas de la función de cada jugador, otras para la jugada mayor para un desempate si es el caso y una “string” para el nombre de dicha mano.
* Para cada jugador: se llama a una función que recibe como parámetros tres matrices y dos enteros, la matriz de las dos cartas del jugador, las cinco cartas de la mesa y la nueva mano del jugador y el numero de la jugada por referencia y la carta mayor para el desempate por referencia.
* Se crea otra matriz más para juntar las 7 cartas.
* Al inicio de cada comprobación de jugada se reiniciarán las matrices por medio de una función, “las7Cartas” y “manoJugador“ estarán vacías de nuevo y “CartasJugador” y “Mesa” serán juntadas de nuevo en “las7Cartas” para hacer una buena comprobación adecuada y además, la variable que determina el número de jugada será -1.
* Se crean dos variables, una de contador y otra para el contador de la jugada mayor.
* Para la escalera real, se abren dos ciclos “for” que recorren la matriz que contiene las 7 cartas, se recorre desde la columna 8 a la 12, si las 5 cartas se encuentran activas en la matriz, se ingresan en la nueva mano del jugador y la jugada del jugador será 1, de lo contrario el contador será igualado en 0.
* Se reinician las matrices y se iguala el contador en 0.
* Para la escalera de color, si la jugada es igual a 1-, se crean dos ciclos “for”, que recorren la matriz, si la carta existe, el contador aumenta en 1, de lo contrario será reiniciado en 0, si el contador llega a un total de 5 la jugada del jugador será 2 y se ingresan las cartas a la mano del jugador, luego se ingresa el valor de la carta mayor.
* Se reinician las matrices y se iguala el contador en 0.
* Para el póker, si la jugada es igual a -1, se abren dos ciclos “for” para recorrer la matriz, si hay 4 cartas iguales en columna se ingresan a la mano, el numero de jugada es 3 y se activa un booleano anteriormente creado para dar paso a la 5 carta, si el booleano es true, se ingresa la primera carta mayor activa encontrada y se agrega para el desempate.
* Se reinician las matrices y se iguala el contador en 0.
* Para el full, si la jugada es igual a -1, se crean dos ciclos “for” para recorrer la matriz por columnas, si la posición es verdadera y las dos siguientes también, se agregarán a la mano del jugador y se activa un booleano para dar paso a la pareja, se vuelve a recorrer la matriz con dos ciclos y si encuentra en su recorrido por columnas a dos cartas iguales, se agregan a la mano del jugador y se activa otro booleano, luego se ingresa la carta mayor para el desempate, si ambos booleanos están activos, la jugada del jugador será 4.
* Se reinician las matrices y se iguala el contador en 0.
* Para el color, si la jugada es igual a -1, se vuelve a recorrer la matriz de la misma manera, si hay 5 cartas del mismo tipo, se ingresan a la mano del jugador y la jugada será 5, la mayor será ingresada a un entero para el desempate.
* Se reinician las matrices y se iguala el contador en 0.
* Para la escalera, si la jugada es igual a -1, se recorre la matriz por columnas, si la carta existe el contador aumenta en 1, si este llega a ser igual a 5, el número de la jugada será 6 y las cartas serán ingresadas a la matriz y el valor carta mayor será ingresada en una variable para el desempate, si no está activo el contador de las columnas aumenta en 1, si este llega a ser 4, el contador principal será 0, todo esto se hace para determinar 5 cartas seguidas sin que se repitan.
* Se reinician las matrices y se iguala el contador en 0.
* Para el trío, si la jugada es igual a -1, se recorre la matriz mediante dos ciclos “for”, si hay tres cartas iguales se ingresan a la mano del jugador y la jugada será la 7, se eliminan de la matriz “las7Cartas” para luego ingresar dos cartas distintas y el valor de la mayor será ingresado en una variable para el desempate.
* Se reinician las matrices y se iguala el contador en 0.
* Para el doble par, si la jugada es igual a -1, se recorre la matriz en dos ciclos “for”, si hay dos cartas iguales se agregan a la mano del jugador y se activa en booleano, se vuelve a recorrer la matriz y se repite el proceso, se agrega la quinta carta de mayor valor y su valor será usado para el desempate y el número de la jugada del jugador será 8.
* Se reinician las matrices y se iguala el contador en 0.
* Para el par, si la jugada es igual a -1, se recorre la matriz buscando un par, si es así se activa un booleano para luego volver a recorrer la matriz y agregar tres cartas más a la mano del jugador, el número de jugada será 9 y la carta mayor de estos tres últimos será usada para el desempate.
* Se reinician las matrices y se iguala el contador en 0.
* Para la carta alta, si la jugada es igual a -1, se toman las 5 cartas más altas y se agregan a la mano del jugador por medio de dos ciclos y el número de jugada del jugador será 10, el valor de la mayor será usado para el desempate.
* Se muestra el nombre del jugador.
* Se muestra el nombre de la mano buscado en una función que retorna el nombre dependiendo del número de jugada.
* Luego de hacer esto con cada uno de los jugadores activos, se busca un ganador por medio de una función.
* Dependiendo de la cantidad de jugadores, se determina el ganador usando el menor valor de la jugada de cada uno, si hay empate, se toma en cuenta la carta mayor obtenida anteriormente, si esta es mayor a la otra en comparación, gana, se muestra en pantalla y se guarda en el historial.

## Determinación de reglas de negocio:

* El juego será de 2 a 4 jugadores.
* Se repartirán dos cartas a cada uno.
* Se pondrán 5 cartas a la mesa y serán comunes para todos los jugadores, una vez entregadas las cartas determinar utilizando las reglas de póker cual es la mejor jugada de cada jugador y quien es el ganador.
* Solo se deben asignar cartas válidas (baraja francesa de 52 cartas), los resultados de cada partida deberán guardarse en un archivo, mismo que podrá consultarse cuando el usuario lo desee, no puede haber empates.