

## Práctica 2 de programación con C++

<b>Asignatura</b>	Fundamentos de programación ( Cóg. 1375)
Profesor responsable de la Asignatura:	Javier Llorente Ayuso
Tipo de actividad:	Actividad de Evaluación Continua (AEC)
Título de la actividad:	Práctica 2 de programación con C++ Práctica 2 de programación con C++

### OBJETIVOS DE APRENDIZAJE

Las competencias específicas a adquirir con esta actividad son:

- Manejar los elementos básicos de programación estructurada.
- Codificar programas y funciones de forma efectiva y eficiente.
- Verificar el buen funcionamiento de un programa o función.
- Documentar programas por medio de los comentarios dentro y fuera del código.
- Utilizar la descomposición modular en C para resolver un problema dividiéndolo en otros subproblemas.
- Seleccionar las estructuras de control apropiadas a un problema dado.
- Seleccionar de los tipos de datos y estructuras de datos apropiadas a un problema dado.
- Realizar análisis básicos en la resolución de problemas algorítmicos.
- Utilizar la capacidad de adaptación de modelos algorítmicos generales en la resolución de problemas concretos.
- Estilo de programación limpio, claro y ordenado.

## INSTRUCCIONES PARA LA REALIZACIÓN Y ENTREGA DE LA ACTIVIDAD

- La **fecha** prevista para la realización de esta Actividad de Evaluación Continua (AEC) se encuentra publicada en el “Foro Convocatoria de Septiembre” así como en el CALENDARIO del Aula Virtual.
- Es **obligatorio** que el **código** enviado se haya **compilado**, ejecutado y probado previamente.
- Recordad que debéis hacer el **código lo más legible posible**, utilizando los comentarios que creáis necesarios, así como los nombres identificadores de las variables más adecuados a cada caso.
- La actividad cumplimentada se envía al profesor a través del  **Buzón de entrega** del Aula Virtual en formato **PDF** con el texto que el alumno estime necesario, **incluyendo el código fuente**; también se entregarán los **ficheros con el código fuente** (.c, .cpp y .h) incluidos en un archivo comprimido (.zip o .rar).
- La **calificación** obtenida, previa corrección y calificación por parte del profesor, se podrá consultar con carácter permanente en el apartado CALIFICACIONES del Aula Virtual.

### **Introducción**

El objetivo de esta actividad, es hacer un “proyecto final”, integrando todos los elementos fundamentales estudiados en la asignatura, como el uso de funciones, los tipos de datos (simples y complejos), así como la metodología de la programación estructurada. Para ello, vais a implementar un pequeño juego de dados.

Es muy conveniente que después de realizar cada parte de un módulo, comprobéis que compile y que funcione. Si no lo hacéis, tendréis el problema de que será muy difícil localizar un error entre todo el código no probado, cuando éste surja.

Para estos ejercicios, necesitáis un entorno de desarrollo donde podáis compilar y ejecutar vuestros programas, al igual que en ejercicios anteriores de esta asignatura. Podéis realizar estos ejercicios con cualquier entorno de desarrollo de C++ que tengáis, siempre que permitan separar el programa en diferentes módulos (.cpp y .h)

### **Enunciado**

El Poker con dados (<https://www.guiaspracticas.com/juegos-de-mesa/poker-con-dados>), es un juego de mesa para dos jugadores, que se juega con 5 dados de Poker. Los dados de Poker son dados de 6 caras que representan los valores de las 6 cartas más altas de la baraja inglesa: As (A), King (K), Queen (Q), jota (J), 10 y 9.

## Desarrollo del juego

El jugador que comienza lanza los 5 dados, una única vez, para simplificar el juego y suma los puntos de la tirada según esta tabla:

Tirada	Puntos
Repoker: 5 cartas iguales.	10
Poker: 4 cartas iguales.	6
Full: un trío y una pareja.	4
Trío: 3 cartas iguales.	3
Dobles parejas: 2 parejas.	2
Pareja: 2 cartas iguales.	1

El orden de los dados no importa, por lo que **no se incluyen escaleras** en el juego.

## Objetivo del juego

El objetivo del juego es conseguir, al final del turno de todos los jugadores, la mano de Poker de mayor valor.

## Final del juego

Una vez que cada jugador ha jugado, quien tenga la combinación más alta, gana. Si ambos jugadores muestran exactamente la misma combinación, el juego es un empate y no ha ganado ninguno de ellos.

La partida puede jugarse a una o más rondas, en este caso podemos asumir que el juego termina una vez que un jugador haya logrado **5 victorias**.

Para ello se deben programar los módulos explicados a continuación.

1) Crea un módulo “dato” con su archivo cabecera (**dado.h**) y su archivo fuente (**dado.cpp**).

En el fichero de cabecera **dado.h** tendremos definido el tipo de datos enumerado que recogerá el resultado de tirar el dado, con los valores siguientes:

```
enum resultadoDado {as=1, king, queen, jota, diez, nueve, otro};
```

El valor **otro** es un valor que se usará excepcionalmente cuando el dado produzca una salida distinta a las anteriores (por un posible error).

También se incluirán las declaraciones de las funciones siguientes (que se implementarán en **dado.cpp**):

```
//Declaración de la Función Tirar Dado  
resultadoDado tirarDado();
```

```
//Declaración de la función mostrarDado  
void mostrarDado(resultadoDado carta);
```

En el fichero **dado.cpp** de este módulo implementaremos dos funciones:

– **resultadoDado tirarDado();**

Simula un dado de poker de 6 caras, que usará la función de números aleatorios **rand ()**:

```
// Saca números aleatorios enteros del 1 al 6  
int numeroDado = 1+ rand ()%6;
```

- **Devuelve** uno de los 6 valores enumerados del tipo **resultadoDado** (**as**, **king**, **queen**, **jota**, **diez**, **nueve**). Para ello la función genera, a partir de un número entero aleatorio **numeroDado** y devolverá el enumerado correspondiente.
- Para implementar esta función se puede usar la función “**rand**” de la librería “**cstdlib**”. También se debe llamar a la función “**srand**” al principio de la ejecución del programa principal tal y como se describe más adelante, para que funcione como un verdadero generador de números aleatorios.

– **void mostrarDado(resultadoDado carta);**

Esta función debe tomar como parámetro un valor enumerado (del tipo **resultadoDado** de los siguientes **as**, **king**, **queen**, **jota**, **diez**, **nueve**) y mostrar por pantalla la cadena de texto correspondiente a cada valor:

“As” para el enumerado **as**

“King” para el enumerado **king**

etc.

**2) Define el módulo “jugador” con su archivos cabecera y fuente, “jugador.h” y “jugador.cpp” respectivamente.**

En el fichero de cabecera (**jugador.h**) definiremos lo siguiente:

- Un tipo estructurado Jugador con los campos `nombre_usuario` (de tipo `string`), `puntos_parciales` (de tipo `int`) y `tiradas_ganadas` (de tipo `int`)
- La declaración de las funciones siguientes:

```
//Función que cargará el nombre del jugador en el campo  
jugador.nombre_usuario
```

```
void solicitarJugador(Jugador& jugador);
```

```
//Función que actualiza el campo jugador.tiradas_ganadas
```

```
void actualizarPartidasGanadas(Jugador& jugador);
```

En el fichero fuente (**jugador.cpp**) se programarán las funciones con el siguiente comportamiento:

- La función `solicitarJugador` pide al usuario por pantalla su nombre “`nombre`” (de tipo `string`) lo almacena en el parámetro (también pasado por referencia “`jugador`”), en el campo `jugador.nombre`:

```
void solicitarJugador(Jugador& jugador);
```

- La función que tiene como parámetros un `jugador` y incrementa en 1 la cantidad de partidas ganadas en el campo `jugador.tiradas_ganadas`

3) Crear un modulo llamado “tirada”, estará formado por dos archivos: el archivo cabecera “tirada.h” y el archivo fuente “tirada.cpp”.

El archivo de cabecera debe incluir lo siguiente **tirada.h**:

```
//Definición del tipo Tirada, como vector de 5 elementos del tipo  
enumerado resultadoDado  
  
typedef resultadoDado Tirada[5];
```

La sentencia anterior se puede utilizar para definir el tipo llamado **Tirada**, que es un vector, de dimensión 5, formado por elementos del tipo enumerado **resultadoDado** (**as**, **king**, **queen**, **jota**, **diez**, **nueve**), de manera que si definimos:

```
Tirada1 tirada;
```

Este tipo de datos almacena valores enumerados que devuelve la función **tirarDado**.

```
// Función que realiza 5 llamadas al dado de poker y muestra por  
pantalla el resultado  
  
int realizarTirada (Tirada& tirada);
```

Esta función se explica en el siguiente apartado, **tirada.cpp**.

El archivo de implementación **tirada.cpp** con la implementación de la función

```
int realizarTirada (Tirada& tirada);
```

Incluirá una instrucción **tirada[i]=tirarDado();** dentro de un bucle que se repita 5 veces para recoger valores del dado (devueltos por la función **tirarDado** de ese módulo Dado) como por ejemplo:

```
tirada[0]=jota , tirada[1]=jota , tirada[2]=king  
tirada[3]=as ,    tirada[4]=queen
```

Además, la función mostrará por pantalla el resultado de cada una de las 5 veces que tiremos el dado utilizando la función **mostrarDado(tirada[i])** del módulo Dado, y devolverá en el parámetro de salida la puntuación de esa tirada a la función principal (main):

Resultado de la tirada:

```
jota  
jota  
King  
As
```

1- **Importante:** como norma general escribiremos los nombres de los tipos de datos que definamos en mayúsculas, diferenciándolos de los nombres de las variables, en minúsculas, siendo por ejemplo **Tirada** el nombre del tipo y **tirada** una variable de ese tipo de datos.

Tanto en este módulo como en los anteriores, recuerda que al terminar cada parte debes comprobar que compila el módulo (en este caso tirada.cpp) y que funciona si se llama a esa función desde el módulo *main.cpp*.

### 3.1. AYUDA: Algoritmo para obtener la cantidad de puntos de una tirada

Para obtener los puntos de cada tirada, de acuerdo a la tabla de la página 3, podemos utilizar el siguiente algoritmo:

#### Versión en Pseudocódigo ejecutable en PSeInt

Algoritmo contadorParejas, con ejemplo de 1 pareja

```
//Dimensionamos el vector para 5 elementos y lo rellenamos para
//probar por ejemplo con queen as as king jota

    Dimension tirada[5]

    tirada[1]="queen"
    tirada[2]="as"
    tirada[3]="as"
    tirada[4]="king"
    tirada[5]="jota"

    //Algoritmo que cuenta cuántas parejas en una misma tirada
    contadorParejas=0;

    Para i<-1 hasta 42 Hacer
        Para j<-i+1 hasta 5 Hacer
            Mostrar Sin Saltar "Ensayo de tirada comparamos dado
            n° ", i, ":" "
            Mostrar tirada[i]," con dado n° ",j,"-",tirada[j]
            Si tirada[i]==tirada[j] Entonces
                mutadora_pareja=Verdadero
                contadorParejas <- contadorParejas + 1
            FinSi
        FinPara
    FinPara
    Mostrar "CONTADOR DE PAREJAS= ",contadorParejas;
FinAlgoritmo
```

2 - Debemos poner este valor para no salir fuera de rango en el vector, al hacer  $j=i+1$  en el siguiente bucle

En el ejemplo anterior el resultado de salida será Ensayo de tirada comparamos dado nº 1: queen con dado nº 2 - as            CONTADOR DE PAREJAS= 1

Mientras que para una tirada como tirada[1]="queen"    tirada[2]="as"    tirada[3]="as"    tirada[4]="king"    tirada[5]="as" el resultado sería    CONTADOR DE PAREJAS= 3

### Versión en C++ del algoritmo

```
int contadorParejas=0;  
for (int i=03; i<4; i++) {  
    for (int j=i+1; j<5; j++) {  
        if (tirada[i]==tirada[j]) {  
            mutadora_pareja=true;  
            contadorParejas++;  
            cout<<"CONTADOR DE PAREJAS "<<contadorParejas<<endl;  
        }  
    }  
}
```

En este algoritmo seguimos la estrategia de comparar cada una de las cinco cartas del vector tirada con las siguientes, de forma que cada vez que encontremos otra carta igual, añadiremos 1 al contador de parejas, pero de manera que no podemos contar una pareja dos veces, es decir, si por ejemplo la tirada[0], es igual que tirada[2]=as, cuando i=0 j=2, contaremos esta pareja una vez, y ya no haremos la comparación con j=2 e i=0.

4 ) En la función “main” del módulo principal controlaremos todo el juego del usuario, con los siguientes pasos:

1. Declara una variable de tipo vector jugador[ ]<sup>4</sup>, del tipo de datos estructura Jugador declarado en el módulo del mismo nombre (en el fichero jugador.h).
2. Declara una variable tiradaActual del tipo de datos estructura Tirada.
3. Pon la siguiente instrucción que permitirá que los números aleatorios sean diferentes en cada ejecución:  
*rand(time(NULL));*  
NOTA: Esta instrucción es complementaria a “rand” y evita que en cada ejecución se produzca la misma secuencia de números aleatorios.
4. Solicita a los jugadores jugador[0] y jugador[1] sus datos con la función “solicitarJugador” y almacénalos en ese vector jugador[ ]

3- Aquí en C++ los índices de los vectores comienzan en 0, por lo que en lugar de 1 a 5 iremos de 0 a 4.

4- Opcionalmente se pueden utilizar dos variables simples jugador0 y jugador1, sin usar vectores, pero esto tendrá un incremento de código que será penalizado en el apartado “Estructuras de datos” de un 20% en ese apartado.

5. Realiza la tirada de cada jugador ( $i=0$ ,  $i=1$ ) devolviendo el resultado de puntos obtenidos desde la función realizarTirada (del módulo Tirada) con:

```
jugador[i].puntos_parciales=realizarTirada(tiradaActual);
```

Mostraremos los puntos obtenidos por cada jugador.

6. A continuación compararemos los puntos\_parciales de cada jugador, si el jugador[0] ha obtenido más puntos que jugador[1] se mostrará por pantalla el texto “El jugador ....” <> “ha ganado” y a continuación mostraremos el total de tiradas ganada por cada jugador.

En caso de empate se mostrará el mensaje “No ha ganado nadie en esta tirada, empatan” y a continuación mostraremos el total de tiradas ganada por cada jugador.

7. Pregunta al usuario anfitrión (el primero de los dos jugadores tiene este rol) si quiere continuar con otra nueva tirada. En caso de que su respuesta sea positiva, se realiza una nueva tirada, siempre que ninguno de los dos jugadores haya alcanzado las 5 tiradas ganadas. En caso de contestar “n” se acaba la ejecución del programa, independientemente de que ninguno de los dos jugadores haya alcanzado las 5 tiradas ganadas.

A continuación se muestran algunos ejemplos de uso del programa.

### **Anexo: Ejemplo de uso del programa.**

JUGADOR 1

Introduzca el nombre: Ana

--> Usuario actual es... Ana

JUGADOR 2

Introduzca el nombre: Pedro

--> Usuario actual es... Pedro

COMENZAMOS LA PARTIDA PARA EL JUGADOR 1: Ana

Resultado de la tirada:

Diez

Jota

King

Nueve

Nueve

Puntos que obtiene el jugador 1 = 1 PUNTOS

COMENZAMOS LA PARTIDA PARA EL JUGADOR 2: Pedro

Resultado de la tirada:

Nueve

Diez

As

Jota

As

Puntos que obtiene el jugador 2 = 1 PUNTOS

No ha ganado nadie en esta tirada, empantan

Ana ha ganado 0 tiradas en total hasta ahora

Pedro ha ganado 0 tiradas en total hasta ahora

Desea seguir jugando (s/n) ?: s

COMENZAMOS LA PARTIDA PARA EL JUGADOR 1: Ana

Resultado de la tirada:

As

Nueve

As

King

Diez

Puntos que obtiene el jugador 1 = 1 PUNTOS

COMENZAMOS LA PARTIDA PARA EL JUGADOR 2: Pedro

Resultado de la tirada:

Diez

Diez

Jota

Jota

Diez

Puntos que obtiene el jugador 2 = 4 PUNTOS

Esta tirada el jugador Pedro ha ganado

Ana ha ganado 0 tiradas en total hasta ahora

Pedro ha ganado 1 tiradas en total hasta ahora

Desea seguir jugando (s/n)?: s

COMENZAMOS LA PARTIDA PARA EL JUGADOR 1: Ana

Resultado de la tirada:

Diez

Queen

King

Queen

Nueve

Puntos que obtiene el jugador 1 = 1 PUNTOS

COMENZAMOS LA PARTIDA PARA EL JUGADOR 2: Pedro

Resultado de la tirada:

Queen

Nueve  
King  
Diez  
Queen  
Puntos que obtiene el jugador 2 = 1 PUNTOS  
No ha ganado nadie en esta tirada, empatan  
Ana ha ganado 0 tiradas en total hasta ahora  
Pedro ha ganado 1 tiradas en total hasta ahora

Desea seguir jugando (s/n)?: s  
COMENZAMOS LA PARTIDA PARA EL JUGADOR 1: Ana

Resultado de la tirada:

As  
King  
Jota  
Queen  
As  
Puntos que obtiene el jugador 1 = 1 PUNTOS  
COMENZAMOS LA PARTIDA PARA EL JUGADOR 2: Pedro

Resultado de la tirada:

Jota  
Nueve  
Jota  
King  
As  
Puntos que obtiene el jugador 2 = 1 PUNTOS  
No ha ganado nadie en esta tirada, empatan  
Ana ha ganado 0 tiradas en total hasta ahora  
Pedro ha ganado 1 tiradas en total hasta ahora

Desea seguir jugando (s/n)?: s  
COMENZAMOS LA PARTIDA PARA EL JUGADOR 1: Ana  
Resultado de la tirada:

Jota

Queen

Diez

King

Jota

Puntos que obtiene el jugador 1 = 1 PUNTOS

COMENZAMOS LA PARTIDA PARA EL JUGADOR 2: Pedro

Resultado de la tirada:

As

Nueve

Nueve

Queen

As

Puntos que obtiene el jugador 2 = 2 PUNTOS

Esta tirada el jugador Pedro ha ganado

Ana ha ganado 0 tiradas en total hasta ahora

Pedro ha ganado 2 tiradas en total hasta ahora

Desea seguir jugando (s/n)?: s

COMENZAMOS LA PARTIDA PARA EL JUGADOR 1: Ana

Resultado de la tirada:

Jota

As

Jota

Nueve

King

Puntos que obtiene el jugador 1 = 1 PUNTOS

COMENZAMOS LA PARTIDA PARA EL JUGADOR 2: Pedro

Resultado de la tirada:

Queen

Nueve

As

King

Jota

Puntos que obtiene el jugador 2 = 4 PUNTOS

Esta tirada el jugador Pedro ha ganado

Ana ha ganado 0 tiradas en total hasta ahora

Pedro ha ganado 3 tiradas en total hasta ahora

Desea seguir jugando (s/n)?: s

COMENZAMOS LA PARTIDA PARA EL JUGADOR 1: Ana

Resultado de la tirada:

King

As

Diez

Diez

As

Puntos que obtiene el jugador 1 = 2 PUNTOS

COMENZAMOS LA PARTIDA PARA EL JUGADOR 2: Pedro

Resultado de la tirada:

Nueve

King

Nueve

As

King

Puntos que obtiene el jugador 2 = 2 PUNTOS

No ha ganado nadie en esta tirada, empatan

Ana ha ganado 0 tiradas en total hasta ahora

Pedro ha ganado 3 tiradas en total hasta ahora

Desea seguir jugando (s/n)?: s

COMENZAMOS LA PARTIDA PARA EL JUGADOR 1: Ana

Resultado de la tirada:

Nueve

King

Jota

King

Queen

Puntos que obtiene el jugador 1 = 1 PUNTOS

COMENZAMOS LA PARTIDA PARA EL JUGADOR 2: Pedro

Resultado de la tirada:

Diez

Queen

As

Jota

Diez

Puntos que obtiene el jugador 2 = 1 PUNTOS

No ha ganado nadie en esta tirada, empatan

Ana ha ganado 0 tiradas en total hasta ahora

Pedro ha ganado 3 tiradas en total hasta ahora

Desea seguir jugando (s/n)?: s

COMENZAMOS LA PARTIDA PARA EL JUGADOR 1: Ana

Resultado de la tirada:

As

Diez

Queen

King

King

Puntos que obtiene el jugador 1 = 1 PUNTOS

COMENZAMOS LA PARTIDA PARA EL JUGADOR 2: Pedro

Resultado de la tirada:

Jota

King

As

Queen

Jota

Puntos que obtiene el jugador 2 = 1 PUNTOS

No ha ganado nadie en esta tirada, empatan

Ana ha ganado 0 tiradas en total hasta ahora

Pedro ha ganado 3 tiradas en total hasta ahora

Desea seguir jugando (s/n)?: s

COMENZAMOS LA PARTIDA PARA EL JUGADOR 1: Ana

Resultado de la tirada:

Diez

Jota

King

Queen

King

Puntos que obtiene el jugador 1 = 1 PUNTOS

COMENZAMOS LA PARTIDA PARA EL JUGADOR 2: Pedro

Resultado de la tirada:

King

Nueve

King

Diez

Jota

Puntos que obtiene el jugador 2 = 1 PUNTOS

No ha ganado nadie en esta tirada, empatan

Ana ha ganado 0 tiradas en total hasta ahora

Pedro ha ganado 3 tiradas en total hasta ahora

Desea seguir jugando (s/n)?: s

COMENZAMOS LA PARTIDA PARA EL JUGADOR 1: Ana

Resultado de la tirada:

Queen

King

Diez

Jota

Jota

Puntos que obtiene el jugador 1 = 1 PUNTOS

COMENZAMOS LA PARTIDA PARA EL JUGADOR 2: Pedro

Resultado de la tirada:

King

King

Nueve

Nueve

Diez

Puntos que obtiene el jugador 2 = 2 PUNTOS

Esta tirada el jugador Pedro ha ganado

Ana ha ganado 0 tiradas en total hasta ahora

Pedro ha ganado 4 tiradas en total hasta ahora

Desea seguir jugando (s/n)?: s

COMENZAMOS LA PARTIDA PARA EL JUGADOR 1: Ana

Resultado de la tirada:

King

Nueve

Queen

King

Nueve

Puntos que obtiene el jugador 1 = 2 PUNTOS

COMENZAMOS LA PARTIDA PARA EL JUGADOR 2: Pedro

Resultado de la tirada:

Queen

Nueve

As

Queen

Nueve

Puntos que obtiene el jugador 2 = 2 PUNTOS

No ha ganado nadie en esta tirada, empatan

Ana ha ganado 0 tiradas en total hasta ahora

Pedro ha ganado 4 tiradas en total hasta ahora

Desea seguir jugando (s/n)?: s

COMENZAMOS LA PARTIDA PARA EL JUGADOR 1: Ana

Resultado de la tirada:

King

As

Queen

Queen

King

Puntos que obtiene el jugador 1 = 2 PUNTOS

COMENZAMOS LA PARTIDA PARA EL JUGADOR 2: Pedro

Resultado de la tirada:

King

King

Diez

As

Nueve

Puntos que obtiene el jugador 2 = 1 PUNTOS

Esta tirada el jugador Pedro ha ganado

Ana ha ganado 1 tiradas en total hasta ahora

Pedro ha ganado 4 tiradas en total hasta ahora

Desea seguir jugando (s/n)?: s

COMENZAMOS LA PARTIDA PARA EL JUGADOR 1: Ana

Resultado de la tirada:

Queen

As

King

Diez

Jota

Puntos que obtiene el jugador 1 = 4 PUNTOS

COMENZAMOS LA PARTIDA PARA EL JUGADOR 2: Pedro

Resultado de la tirada:

Diez

Jota  
Diez  
King  
Queen  
Puntos que obtiene el jugador 2 = 1 PUNTOS  
Esta tirada el jugador Pedro ha ganado  
Ana ha ganado 2 tiradas en total hasta ahora  
Pedro ha ganado 4 tiradas en total hasta ahora

Desea seguir jugando (s/n)?: s  
COMENZAMOS LA PARTIDA PARA EL JUGADOR 1: Ana

Resultado de la tirada:

As  
Queen  
As  
King  
Jota  
Puntos que obtiene el jugador 1 = 1 PUNTOS  
COMENZAMOS LA PARTIDA PARA EL JUGADOR 2: Pedro

Resultado de la tirada:

Nueve  
Jota  
As  
Jota  
Nueve  
Puntos que obtiene el jugador 2 = 2 PUNTOS  
Esta tirada el jugador Pedro ha ganado  
Ana ha ganado 2 tiradas en total hasta ahora  
Pedro ha ganado 5 tiradas en total hasta ahora

**El jugador Pedro ha alcanzado las 5 tiradas y HA GANADO EL JUEGO FINAL**