

Universidad Rafael Landívar
Facultad de Ingeniería
Ingeniería Química
Introducción a la Programación – Sección 07
Catedrático: Ing. Edwin Timoteo Chocoy Cordón

PROYECTO PRÁCTICO No. 02

María Alexandra Sánchez Vargas – 1055323

Guatemala, 12 de noviembre de 2023

INDICE

INTRODUCCIÓN	1
ANÁLISIS	2
ENTRADAS	2
PROCESOS.....	2
SALIDAS	4
RESTRICCIONES.....	4
DISEÑO	5
DIAGRAMA DE FLUJO	5
DIAGRAMA DE CLASES	6
CONCLUSIONES	7
RECOMENDACIONES	8
REFERENCIAS.....	9
LIBRERIAS UTILIZADAS Y SU UTILIZACIÓN	9
REFERENCIAS BIBLIOGRÁFICAS	9
ANEXOS	10
MANUAL DE USUARIO.....	10

INTRODUCCIÓN

Para el laboratorio de Introducción a la Programación, se realizó un programa como segundo proyecto, en el cual se implementa un juego de dados simple. El juego es una competencia de puntos entre el jugador, el cual es el usuario, y la casa. Este juego se basa en lanzar dos dados aleatoriamente por cada lanzamiento, pero el jugador puede elegir cuantos lanzamientos hacer por la cantidad de partidas que el también desee. El jugador gana puntos si la suma de sus dados es igual a 11, 12, 6 o 3. La casa gana puntos si la suma de sus dados es igual a 4, 6 o 10. Si la suma de los dados es igual a 2, 3, 5, 7, 8 o 9, ambos jugadores ganan puntos. Si llegara a existir algún desconocimiento acerca del juego, de las reglas o del programa, se elaboró un manual de usuario, para facilitar la experiencia del usuario "jugador".

El código utiliza una estructura "for" para iterar sobre los lanzamientos de dados, que es una herramienta eficiente para iterar sobre una secuencia de elementos. El código también utiliza condicionales "if" para verificar si la suma de los dados cumple alguna de las condiciones de victoria mencionadas con anterioridad, que es correcto, ya que las reglas del juego se basan en la suma de los dados. El código también está diseñado para calcular el porcentaje de victorias del jugador por partida, para que este pueda evaluar su desempeño. Por último, el programa le enseña al usuario cada fin de partida, sus resultados, tanto como su porcentaje, el ganador de la partida, resultados finales, numero de lanzamientos iguales o distintos y la cantidad de lanzamientos en el cual el gana puntos.

Para el desarrollo de este programa, se utilizó la herramienta de Visual Studio la cual se utilizó a lo largo del segundo ciclo, esta aplicación es un entorno de desarrollo integrado desarrollado por Microsoft. Visual Studio incluye una amplia gama de herramientas y funcionalidades para poder editar, depurar, compilar, probar y ejecutar diversos códigos.

ANÁLISIS

ENTRADAS

IMAGEN No. 01

```
6 static void Main(string[] args)
7 {
8     Random aleatorio = new Random();
9     Console.WriteLine("PROYECTO No. 02");
10    Console.WriteLine("BIENVENIDO A UN JUEGO DE DADOS ");
11    Console.WriteLine();
12    Console.Write("Ingrese el numero de partidas: ");
13    int partidas = Convert.ToInt32(Console.ReadLine());
14    Console.Write("Ingrese el numero de lanzamientos por partida: ");
15    int lanzamientos = Convert.ToInt32(Console.ReadLine());
16    Console.WriteLine();
```

En la imagen se pueden observar dos entradas, entradas son los datos que el usuario proporciona al código para que este lo procese. En este código se le pide al usuario que ingrese dos variables de números enteros. La primera es el número de partidas que el usuario desee jugar y el segundo es la cantidad de lanzamientos de dados que va a haber por cada partida.

PROCESOS

IMAGEN No. 02

```
28 for (int i = 1; i <= lanzamientos; i++)
29 {
30     int dado1 = aleatorio.Next(1, 7);
31     int dado2 = aleatorio.Next(1, 7);
32     int sumaDados = dado1 + dado2;
```

En la imagen se puede observar cómo se declararon dos variables aleatorias, estos corresponden a cada uno de los dos dados. La cantidad de números que son generados por los dados dependen a la cantidad de lanzamientos ingresados anteriormente. Para este proceso se utilizó un "for", el cual se utiliza para iterar sobre una secuencia de elementos. Por último, en esta imagen, también se realizó la suma de ambos dados para futuros procesos.

IMAGEN No. 03

```
34 if (dado1 == dado2)
35     lanzamientosIguales++;
36 else
37     lanzamientosDiferentes++;
```

En la imagen se observa como el código, por medio de un “if”, analiza si los dos dados dan el mismo valor, se cuentan como lanzamientos iguales, mientras que, si no son iguales, el programa los cuenta como lanzamientos diferentes.

IMAGEN No. 04

```

39  if (sumaDados == 11)
40  {
41      if (puntosJugador == 0)
42          puntosCasa = puntosCasa + 6;
43      else
44      {
45          puntosJugador = puntosJugador + sumaDados;
46          victoriasJugador = victoriasJugador + 1;
47      }
48  }
49  else if (numeroLanzamiento == 1 && (sumaDados == 12 || sumaDados == 6))
50  {
51      puntosJugador = puntosJugador + 12;
52      victoriasJugador = victoriasJugador + 1;
53  }
54  else if (numeroLanzamiento == 1 && (sumaDados == 4 || sumaDados == 6 || sumaDados == 10))
55      puntosCasa = puntosCasa + 12;
56  else if (sumaDados == 2 || sumaDados == 3 || sumaDados == 5 || sumaDados == 7 || sumaDados == 8 || sumaDados == 9)
57  {
58      puntosJugador = puntosJugador + sumaDados;
59      victoriasJugador = victoriasJugador + 1;
60  }
61  else
62      puntosCasa = puntosCasa + sumaDados;

```

En esta imagen, se utilizó la condición “if” varias veces para declarar, que, si la suma de los dados es igual a 11, la casa gana 6 puntos. Si la suma de los dados es 12 o 6 en el primero tiro, al jugador se le es sumado 12 puntos. Si la suma de los dados es 4, 6 o 10 del primer lanzamiento del jugador, la casa gana 12 puntos. Si la suma de los dados es 2, 3, 5, 7, 8, o 9, la suma es el punteo del jugador o de la casa. Por último, si la suma de los dados da 11 antes de que el jugador haya ganado algún punto, la casa gana 6 puntos.

IMAGEN No. 05

```

70  double porcentajeVictoria = Math.Round(((double)victoriasJugador / lanzamientos) * 100, 2);
71  string mensajeFinal;
72
73  if (puntosJugador == puntosCasa)
74  {
75      mensajeFinal = "Empate";
76  }
77  else if (puntosJugador > puntosCasa)
78  {
79      mensajeFinal = "El ganador es el jugador";
80  }
81  else
82  {
83      mensajeFinal = "El ganador es la casa";
84  }

```

En esta imagen, se observa como el algoritmo crea el porcentaje de victoria del jugador, para esto divide los lanzamientos de la partida entre las victorias que tuvo el jugador, todo esto multiplicándolo por 100. Mas abajo se observa la variable del mensaje final, este mensaje final es de tipo “string” y declara quien fue el ganador de la partida. Utilizando la condición if, si el punteo del jugador es igual al de la casa, es un empate, si el punteo del jugador es mayor, gana él y si no cumple ninguno de los dos casos anteriores, la casa gana.

SALIDAS

IMAGEN No. 06

```
86 Console.WriteLine();
87 Console.WriteLine("- PARTIDA No. 0" + j + " -");
88 Console.WriteLine();
89 Console.WriteLine("Ganador: " + mensajeFinal);
90 Console.WriteLine("Lanzamiento de dados: ");
91 foreach (int d in lanzamientoDados)
92 {
93     Console.Write(d + " ");
94 }
95 Console.WriteLine();
96 Console.WriteLine("Porcentaje de victoria del jugador: " + porcentajeVictoria + "%");
97 Console.WriteLine("El jugador gano puntos en " + victoriasJugador + " lanzamientos");
98 Console.WriteLine("El jugador obtuvo numeros diferentes en " + lanzamientosDiferentes + " lanzamientos");
99 Console.WriteLine("El jugador obtuvo numeros iguales en " + lanzamientosIguales + " lanzamientos");
100 Console.WriteLine("Resultados: PuntosJugador = " + puntosJugador + ", PuntosCasa = " + puntosCasa);
101
102 Console.ReadKey();
```

En la imagen se observa cómo es que por partida, se le muestra en pantalla al usuario el número de partida jugada, el mensaje final que indica cual fue el ganador, los lanzamientos de dados que tuvo el jugador en la partida, el porcentaje de victoria que tenía el jugador para ganar, la cantidad de lanzamientos en la que el jugador gano puntos, el número de lanzamiento en el que el jugador tubo números diferentes e iguales y el resultado final de la partida, es decir, la puntuación de cada uno.

RESTRICCIONES

IMAGEN No. 07

```
12 Console.Write("Ingrese el numero de partidas: ");
13 int partidas = Convert.ToInt32(Console.ReadLine());
14 Console.Write("Ingrese el numero de lanzamientos por partida: ");
15 int lanzamientos = Convert.ToInt32(Console.ReadLine());
```

Las restricciones que existen en el código, es que cuando el usuario ingresa el número de partidas y el número de lanzamientos por partidas, estos dos valores deben ser números enteros positivos para que el programa funcione.

DISEÑO

DIAGRAMA DE FLUJO

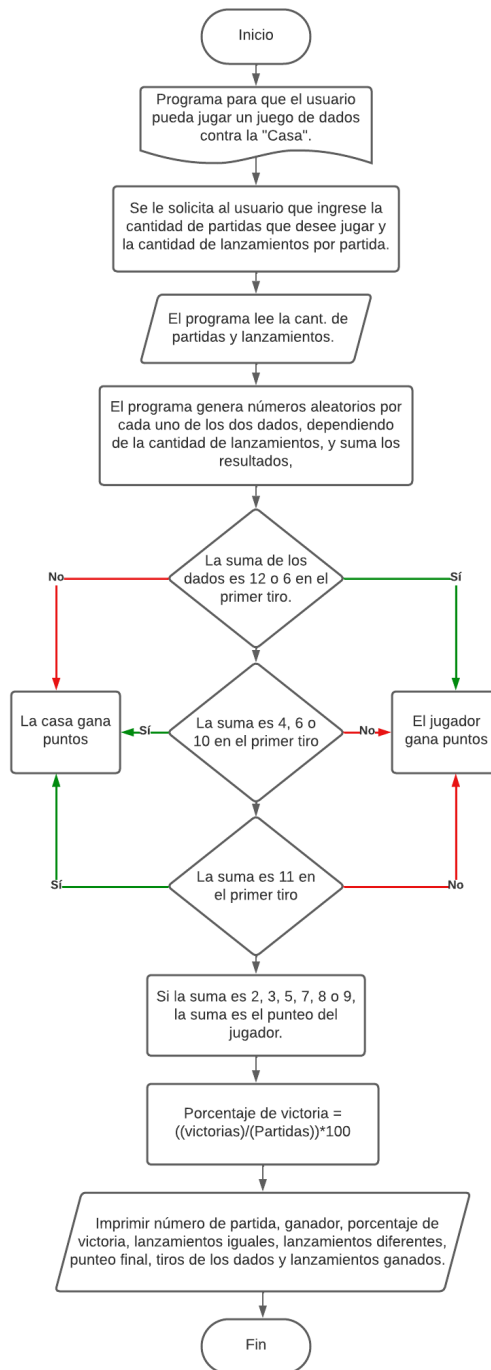


DIAGRAMA DE CLASES

Clase
Dados
Atributos
int partidas
int lanzamientos
Int numeroLanzamiento
Int puntosJugador
Int puntosCasa
Int lanzamientoDados
Int lanzamientosDiferentes
Int lanzamientosIguales
Int victorias Jugador
String mensajeFinal
Métodos
Double pocentajeVictoria
String mensajeFinal
Int dado1
Int dado 2
Int sumaDados

CONCLUSIONES

1. El uso de un bucle "for" para iterar sobre los lanzamientos de dados ayuda para una elección eficiente en el diseño del programa. La estructura de control "for" es especialmente útil cuando se conoce de antemano la cantidad de iteraciones necesarias, como en este caso, cuando el jugador decide cuántos lanzamientos desea realizar por partida.
2. Se implementaron las reglas del juego utilizando las condicionales "if" para facilitar de manera clara y comprensible el desarrollo del programa. Las reglas del juego están bien definidas y se reflejan directamente en el código, lo que agiliza su comprensión para el usuario.
3. La presentación al final de cada partida, que incluye resultados, porcentaje, ganador, número de lanzamientos y otros datos relevantes, contribuye a una experiencia completa para el usuario. Proporcionando una retroalimentación detallada y transparente sobre su desempeño, lo que es esencial para mantener el interés y la participación.

RECOMENDACIONES

1. Es recomendable que, a la hora de describir un código, se haga de forma clara y organizada. Para esto, es importante seguir un orden durante la programación, como lo puede hacer declarar las entradas, realizar todos los procesos y de ultimo dejar todas las salidas. Esto ayudara a mantenerlo limpio y que sea más fácil de leer o modificar para uno u otras personas.
2. Para la elección de condicionales, procesos, funciones, entre otras herramientas disponibles, es necesario saber qué es lo que se desea que realice el programa, las instrucciones o reglas en los que se tiene que encasillar, pues cada herramienta tiene una utilidad diferente y la función de ellas no es complicar el programa, sino buscar soluciones sencillas a los problemas.
3. A la hora de pedirle al usuario que ingrese un dato, es importante ser bastante específicos con el tipo de dato que se desea. El programa puede almacenar el dato como un texto, un decimal, un numero entero, un booleano, entre otros. Por lo que, si se ingresa el dato erróneo, el programa para de funcionar.

REFERENCIAS

LIBRERIAS UTILIZADAS Y SU UTILIZACIÓN

```
1 using System;
```

El using System es una instrucción importante que se debe usar en todos los archivos de código C#. Esta instrucción facilita el acceso a las clases y métodos fundamentales del lenguaje de programación C#.

REFERENCIAS BIBLIOGRÁFICAS

EDraw. (2023). *Cómo crear un Diagrama de Flujo*. Wondershare.
<https://www.edrawsoft.com/es/create-programming-flowchart.html>

ANEXOS

MANUAL DE USUARIO

1. El usuario debe ingresar al programa en el caso que quiera participar a un juego de dados entre el jugador y la "Casa".

```
PROYECTO No. 02  
BIENVENIDO A UN JUEGO DE DADOS
```

2. El usuario como jugador debe ingresar el número de partidas que desee jugar contra la "Casa".

```
Ingrese el numero de partidas: |
```

3. El usuario como jugador debe ingresar el número de lanzamientos de dados que quiere que haya por partida.

```
Ingrese el numero de lanzamientos por partida: |
```

4. A la hora de los lanzamientos, el jugador debe conocer las reglas del juego, estas son:
 - Si la suma de los dados es 12 o 6 en el primer tiro, el jugador gana 12 puntos
 - Si la suma es 4, 6 o 10 de la primera tirada el jugador pierde y la "Casa gana" 12 puntos.
 - Si la suma es 2, 3, 5, 7, 8 o 9 en el tiro, la suma es el punteo del jugador o la "Casa".
 - Un jugador puede perder si la suma tira un 11 antes de haber ganado ningún punto, para este caso la "Casa" gana 6 puntos.
5. El programa lanza los dados por el jugador de manera aleatoria, y por partida le enseña sus resultados.

```
- PARTIDA No. 01 -
```

```
Ganador: El ganador es el jugador  
Lanzamiento de dados:  
2 5 2 5 3 5  
Porcentaje de victoria del jugador: 100%  
El jugador gano puntos en 3 lanzamientos  
El jugador obtuvo numeros diferentes en 3 lanzamientos  
El jugador obtuvo numeros iguales en 0 lanzamientos  
Resultados: PuntosJugador = 22, PuntosCasa = 0
```

6. El juego termina.

¡MUCHAS GRACIAS POR JUGAR!