



## PROYECTO PRÁCTICO NO. 02

### Dados

Hay muchos juegos populares de dados, uno de ellos es el siguiente, en el cual el jugador tiene dos dados, cada dado tiene 6 caras, y cada una tiene un número 1,2,3,4,5,6. Al lanzar los dados la suma de ambos resultados implica diferentes reglas, las cuales 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.

Para ganar se decide la cantidad de partidas o tiradas de dados y al llegar a la cantidad el jugador gana si tiene más puntos que la "Casa" o sino "La casa gana"

Se le ha solicitado simular una cantidad N de partidas de M tiros cada una, dicha cantidad debe ser solicitada al usuario al iniciar, guardando los resultados de cada partida

Hay dos formas de realizar la simulación, una es por un archivo de texto de entrada que tendrá todas las jugadas y otra es que los valores de los tiros se generen de forma aleatoria.

En el caso del archivo de texto, se debe verificar que tenga la cantidad de tiros indicados por el usuario, se debe solicitar el nombre del archivo y su ubicación al iniciar, el archivo debe tener las jugadas en el siguiente formato:

```
2 4 6 5 2 2 6 6 8 6 4 6
2 4 3 3 4 4 8 6 4 6 3 6
6 4 6 4 2 4 1 6 5 6 1 6
```

En este caso se asume que el usuario ingreso que la cantidad de partidas son 3 y de tiros son 6. En el archivo se debe validar que los números sean válidos y que la cantidad de tiros y partidas venga correcta, mostrando mensajes de error en caso contrario. Cada fila representa una partida y cada dos números representa un tiro.

En el caso de las jugadas aleatorias se deben generar los tiros y partidas automáticamente según lo ingresado por el usuario y que los números aleatorios correspondan a los indicados.



Al finalizar se debe mostrar la siguiente información:

1. El ganador de la partida: El jugador o la casa.
2. Los tiros realizados en cada partida, cada dado por separado.
3. Cuántos tiros ganó el jugador, es decir en cuantos tiros el ganó puntos.
4. Cuál era su probabilidad de ganar.
5. En cuántos tiros le salieron números pares e impares.
6. En cuántos tiros le salieron números iguales.
7. Punteo final.

### Aspectos por evaluar

- Validación de errores.
- Adecuada aplicación de los conocimientos.
- Calidad de la documentación: ortografía, orden, limpieza y que esté completa.
  - Carátula
  - Introducción
  - Análisis
  - Diseño
    - Diagrama de Flujo
    - Diagrama de Clases
  - Conclusiones
  - Recomendaciones
  - Referencias
    - Librerías utilizadas y su utilización
  - Anexos
    - Manual de Usuario
- Calidad de la solución propuesta: que solucione el problema (que haga lo que requiere el sistema de reserva) en forma eficaz.
- Funcionalidad del programa: debe cumplir a cabalidad con todos los requerimientos.
- **Evidencia de la creación del programa y dominio de los conceptos utilizados.**
- Creatividad.

### Ponderación

Documentación	10
Análisis	10
Diseño	10
Codificación	70
Total	100



## Consideraciones

- Se trabajará en modalidad individual.
- Se debe realizar una solución utilizando la Consola, en lenguaje de programación C#.
- Toda solución presentada debe compilar correctamente para poder tener derecho a revisión.
- La utilización de código de terceros debe estar completamente documentado, referenciado y justificada su utilización, debe demostrarse el dominio completo de lo implementado.
- **Se podrá demandar que en la calificación presencial del proyecto se realicen cambios de funcionalidad.**