

Laboratorio No.1
Análisis

- **Requisitos funcionales**
 - Tira un número “random”
 - Va sumando el valor de la cara de dados
 - Si uno de los dados tira 1, se reinician los dados
 - Al llegar a 100 puntos, el jugador gana
 - El jugador decide si, después de tirar, se vuelve a arriesgar a tirar otra vez
 - El jugador debe parar de lanzar los dados al momento de obtener 20 o más puntos en una ronda
- **Propósito de las clases**
 - Dado: contiene el número máximo y mínimo de los dados, tira un número random
 - Jugador: le da opciones de empezar juego, tirar dados y continuar juego, compara resultados y los suma (-1, +100)
- **Propósito de cada atributo**
 - suma#: suma los puntos tirados en los dados de cada jugador
 - valor#: guarda valores de dados
 - dado#: da cantidad de dados a usar
 - reinicia: reinicia la cuenta a 0
 - continuar: continúa juego
 - num: valor random de dado
 - jugadorActual: jugador que efectúa la acción en ese momento
 - pts: puntaje acumulado
- **Propósito de cada método de cada clase**
 - setsuma#: setter para contador de puntos
 - getsuma#: obtiene las sumas
 - setvalor#: setter para valor de dados
 - getvalor#: obtiene los valores del dado y los muestra
 - set pts: setter para puntaje obtenido y acumulado
 - getpts: obtiene puntaje actual
 - setreinicia#: setter para reiniciar contador a 0
 - getreinicia#: obtiene el contador y lo reinicia a 0
 - setjugador#: setter para cada jugador
 - getjugador#: obtiene al jugador y hace lo que requiere