



## INTRODUCCIÓN A LA COMPUTACION

A continuación Ud. encontrará un problema que debe ser resuelto por equipos de no más de dos estudiantes, la solución final debe ser implementada en un programa escrito en lenguaje Python, utilizando funciones. La solución del problemas junto a su informe debe ser subida a la plataforma virtual del curso el día Miércoles 04 de Diciembre hasta las 23:50 P.M., no se recibirán trabajos después de la fecha y hora indicada. La no presentación del trabajo será evaluada con 1.0.

### Problema

El juego del Gato, se juega entre dos personas y consiste en que alternadamente cada uno de los jugadores hace una jugada en alguno de los casilleros disponibles, intentando ser el primero en completar con sus fichas una línea de tres casilleros.

El objetivo de esta proyecto es que Ud. construya un programa que implemente en lenguaje Python 3 el juego del Gato, de forma tal que uno de los jugadores sea una persona, la que ejecuta el programa, y que el otro jugador sea el computador.

Se dispone de 9 posiciones ordenadas en forma rectangular como indica la figura:

1	2	3
4	5	6
7	8	9

El juego comienza cuando los jugadores lanzan un dado para determinar quién juega primero. El jugador que obtenga el valor más alto debe comenzar.

Al comienzo del juego las 9 posiciones están vacías. El jugador que inicia el juego selecciona una casilla vacía y la marca. A continuación el segundo jugador selecciona una casilla vacía y la marca. Enseguida juega nuevamente el primer jugador, marcando alguna casilla vacía. Esto continua hasta que alguno de los dos jugadores marca 3 casillas que estén en línea (puede ser en diagonal), si ninguno de los jugadores logra este objetivo y no hay más casillas vacías, el juego se declara en empate.

### Ejemplo:

Lanza dado Humano -> 3

Lanza dado Computador -> 5

Inicia el Juego: Computador:

Computador: Juega en 1			Humano: Juega en 3			Computador: Juega en 5			Humano: Juega en 6			Computador: Juega en 9		
	C			C	H	C		H	C		H	C		H

El Ganador es: Computador.

Desea Jugar Nuevamente ( 0 = Si / 1 = No): 0

### Otro Ejemplo

Lanza dado Humano -> 6

Lanza dado Computador -> 2

Inicia el Juego: Humano:

Humano: Juega en 3			Computador: Juega en 5			Humano: Juega en 6			Computador: Juega en 9			Humano: Juega en 1		
		H			H			H			H	H		H

Computador: Juega en 2	Humano: Juega en 8	Computador: Juega en 4	Humano: Juega en 7	
H C H C H C	H C H C H H C	H C H C C H H C	H C H C C H H H C	

**Se ha producido un empate.**

Desea Jugar Nuevamente ( 0 = Si / 1 = No): 1

El programa desarrollado por usted debe permitir el juego entre un humano y el computador. Para ello, el programa debería tener una doble funcionalidad: jugador y administrador del juego. Como administrador del juego, el programa debe ser capaz de:

- Simular el lanzamiento del dado, tanto para el humano como para el computador
- Desplegar el tablero con las jugadas cada vez que alguno de los jugadores realice su movimiento
- Indicar el ganador o si corresponde el empate.
- Dar la posibilidad de seguir jugando.

El computador debe realizar sus jugadas, considerando:

- Determinar los lugares en los que puede jugar (no jugados)
- Buscar una posición adecuada para jugar, esto es, debe determinar si hay alguna casilla que al marcarla gane el juego.
- Si no hay una casilla con las características anteriores, debe jugar en forma aleatoria en alguna casilla disponible.

El programa debe mostrar una interfaz similar al ejemplo.

Cada vez que un usuario comience un juego, ya sea porque inicie el programa o porque decida, sin salir del programa, el programa debe determinar que jugador comienza.

**Su trabajo será evaluado de acuerdo a la siguiente Rubrica:**

	1	3	5	7	Total
Revisión del Código.  <b>(20%)</b>	Código desordenado, sin uso de las normas de programación, no utiliza comentarios, nombres de variables inadecuados. No utiliza funciones.	Código Ordenado Usa identificadores de variables adecuado. Incorpora comentarios mínimos o existe sobresaturación de comentarios. Uso básico de funciones.	Código ordenado. Uso adecuado de variables (listas, string, etc.). Uso adecuado de estructuras algorítmicas Uso eficiente de los comentarios. Uso adecuado de funciones	Cumple con todas las normas de programación.	
Solución Computacional.  <b>(40%)</b>	No se ejecuta. Errores de Ejecución. Programa cae en Loop.	Ejecuta pero con errores de lógica. Ejecuta pero no llega a la solución.	Llega a una solución parcial. Tarda más de un minuto en llegar a la solución. No cumple con el formato de salida.	Llega a la solución en menos de un minuto cumpliendo exactamente con el formato de salida.	
Contenido del informe  <b>(20%)</b>	No cumple con todos los apartados del informe o estos están	Cumple con todos los apartado del informe, pero están	Cumple con todos los apartado del informe, están desarrollados	Cumple con todos los apartado del informe, están desarrollados	

	no expresan lo solicitado.	desarrollados pobemente.	acuerdo a lo solicitado, pero hay desprolijidad en su presentación.	de acuerdo a lo solicitado y se presenta de Manera prolja.	
Gramática y ortografía  <b>(20%)</b>	El informe contiene 9 o más errores gramaticales u ortográficos.	El informe contiene entre 5 y 8 (inclusive) errores gramaticales u ortográficos.	El informe contiene entre 1 y 4 (inclusive) errores gramaticales u ortográficos.	El informe no contiene ningún error grammatical u ortográfico.	

## CONTENIDO DEL INFORME

EL proyecto debe ser subido a la plataforma, en el hora y fecha indicada, en un archivo comprimido (zip o rar) que contenga sólo el código fuente del programa (formato .py) y el informe en formato PDF.

El informe debe considerar mínimo: la estrategia de solución empleada, la descripción de las estructuras de datos utilizadas, los alcances y limitaciones del programa. También debe tener un apartado con el manual de usuario.

El formato a seguir por el documento es la siguiente: Primera hoja, identificación de la universidad, departamento, carrera, asignatura, nombre de los profesores, nombre de los estudiantes y fecha de entrega.

**A Continuación debe venir la introducción, diseño de la solución, detalles de la implementación, limitaciones y finalmente el manual de usuario.**

**Introducción:** Contiene la descripción general del problema (no el enunciado), en este punto debe quedar claro cuál es el problema que se debe resolver.

**Diseño de la solución:** Descripción clara de la manera como se abordó el problema.

Debe explicar cómo el mundo real ha sido modelado en la solución planteada. A

continuación se deben describir los algoritmos y estructuras de datos principales que transforman el mundo en un modelo informático utilizado para alcanzar la solución. Como usted ya sabe, un problema debe ser descompuesto en subproblemas, es en este punto donde se debe indicar cuales fueron los subproblemas planteados y como a partir de ellos se llega a la solución general del problema.

**Detalles de implementación:** Esto se refiere a la transformación de los algoritmos en programas escritos en lenguaje Python 3, se debe incluir cualquier detalle que usted considere relevante.

**Limitaciones:** Se deben indicar las limitaciones del programa, las cuales pueden ser propias del enunciado o propias de la solución planteada. Aquí deben quedar claramente registradas aquellas cosas que usted no resolvió y que son fuente de error de su programa. Como por ejemplo errores que no pudo corregir por falta de tiempo, en cuyo caso debe detallar cual sería la estrategia de solución.

**Manual de Usuario:** Permite que cualquier usuario interactúe con el programa, sin tener conocimiento del interior del programa. Debe contener todos los detalles que el usuario debe conocer para poder utilizar el programa. Generalmente se nutre de imágenes que muestran ejemplos de ejecución.