

# Evaluación N°3

## Algoritmos y programación

Anggelo Urso G.  
anggelo.urso@inacapmail.cl

30 de junio de 2020

### 1 Introducción

El presente trabajo tiene por finalidad evaluar los contenidos vistos en clases de la unidad 4, a través del desarrollo de un programa en lenguaje de programación C, considerando los tópicos vistos:

- Entrada y salida de datos.
- Sentencias de control de flujo:
  - IF - ELSE
  - WHILE
  - FOR
- Funciones

Con esta evaluación, se persigue que los alumnos demuestren sus capacidades de desarrollo de código a través de un juego tradicional.

En el presente documento, se detallan las reglas y el problema a ser abordado por los alumnos. En la primera parte de éste, se realiza la descripción del problema y sus reglas.

En la segunda parte del documento se especifican las restricciones del problema, así como las restricciones de la evaluación. Y en la última sección se detallan las fechas y formatos.

El trabajo podrá ser realizado en equipos de dos personas, las cuales no necesariamente podrán ser las mismas de la evaluación previa (evaluación 2) o realizado de forma individual. Las fechas, formatos y restricciones se encuentran descritas en el apartado **fecha de entrega y formato**.

## 2 Descripción del problema

Dado a que el tradicional juego del piedra/papel/tijera presenta una cantidad significativa de posibles empates, existe una variante bastante interesante para aumentar el número de combinaciones y así reducir la cantidad de empates que se puedan generar.

El juego se llama piedra/papel/tijera/lagarto/spock<sup>1</sup>. La idea de la actividad es que logren programar el juego, utilizando para ello el lenguaje de programación en C. Éste podrá ser jugado a través de las siguientes modalidades:

1. Jugador vs Computador
2. Jugador vs Jugador

El alumno deberá programar el juego, entregando las opciones correspondientes al jugador o los jugadores y evaluar: quien ha sido el ganador, quien pierde o si hubo empate.

## 3 Reglas del juego

Tal como el tradicional juego piedra/papel/tijera, dependiendo de la opción se evalúan los resultados de acuerdo a la siguiente tabla.

Jugador 1 \ Jugador 2	Piedra	Papel	Tijera	Lagarto	Spock
Piedra	E	-	X	X	-
Papel	X	E	-	-	X
Tijera	-	X	E	X	-
Lagarto	-	X	-	E	X
Spock	X	-	X	-	E

En esta tabla se puede observar las combinaciones de empates (**E**), cuando el jugador 1 gana (**X**) y cuando el jugador 2 gana (-). Cuando se juega contra la computadora, el jugador escoge una de las alternativas y el computador escoge otra de forma aleatoria.

Para simplificar el problema se utilizarán tiradas secuenciales, es decir, el jugador 1 escogerá su opción y luego el jugador 2. En el caso de jugar contra la computadora, el jugador 1 realizará su tirada y el computador escogerá una de las alternativas disponibles de forma aleatoria.

Al final de la tirada, se deberá desplegar quien es el jugador que gana (o si gana la computadora). Si se juega contra la computadora, el programa deberá indicar cuál es la tirada que escogió la computadora e indicar el resultado.

<sup>1</sup>Agradecimientos a The Big Bang Theory, capítulo donde se explica el juego <https://www.youtube.com/watch?v=43ijRL7w8z8>

## 4 Restricciones del problema

Como restricciones del juego, deberán considerar que las opciones a ser ingresadas deberán presentarse en un menú de opciones a los jugadores, cada uno de ellos escogerá una de las 5 opciones, o decidirán volver a la pantalla anterior. Para esto podrán utilizar la siguiente imagen como referencia:

```
Seleccione una tirada:

1.- Piedra
2.- Papel
3.- Tijera
4.- Lagarto
5.- Spock
6.- Volver

Tirada Jugador 1: 1
```

Si observan, cada una de las opciones está mapeada con un número, incluyendo la opción volver, que redirige a cualquiera de los dos jugadores a la pantalla inicial:

```
*****
*                               *
*           Juego               *
*  Piedra, papel, tijera       *
*   lagarto, spock            *
*                               *
*****

Seleccione opción de juego:

1.- Jugador vs PC
2.- Jugador vs Jugador
3.- Salir

Modo juego: 1
```

Acá si observan, tienen las tres opciones: Jugador vs computadora, jugador vs jugador y salir; la cual termina la ejecución del programa. Como datos de entrada deberán ingresar un número de dicho menú para ir navegando.

Otra de las restricciones, es la utilización de la función random. Para eso deberán incluir el siguiente código:

```
1 #include <stdio.h> // para usar printf y scanf
2 #include <stdlib.h> // para usar rand y srand
3 #include <unistd.h> // para getpid()
4
5 int main() {
6     srand(getpid()); // genera una semilla aleatoria
7     int numero = rand() % 5; // genera un random en el anillo
8     return 0;
9 }
```

La función `srand(getpid())` puede en cualquier parte del código, pero recomiendo que la dejen en el `main` del programa, así la función `rand()` tendrá acceso a la semilla aleatoria en cualquier parte del programa que lo definan.

El programa no deberá terminar hasta que no se seleccione la opción 3 del menú de juego (Salir). Hasta antes de eso, el jugador podrá ir seleccionando distintas modalidades de juego.

La cabecera presentada en la imagen 4, es meramente referencial y no es necesario dejar una cabecera similar o incluirla en el desarrollo de la actividad.

## 5 Restricciones de la evaluación

Para la presente evaluación y en línea con los objetivos planteados al comienzo del documento, se espera que los alumnos entreguen un trabajo con las siguientes restricciones:

1. Se deberá entregar un archivo fuente (formato `.c`) que permita ejecutar su programa.
2. Deberá respetar el formato de entrada de datos (solo enteros).
3. Deberá contar con a lo menos 1 función y a lo menos 2 ciclos.
4. Deberán necesariamente utilizar las librerías descritas en el segmento de código 4.
5. Deberán utilizar las funciones `rand` para la tirada de la computadora, de manera que sea una elección aleatoria.
6. La compilación no deberá proporcionar ningún tipo de error o advertencia.
7. No es obligatorio que los mismos alumnos que entregaron la evaluación 2 como equipo, entreguen la evaluación 3. Los equipos pueden mutar o pueden decidir hacer la actividad de forma individual.

## 6 Fecha de entrega y formato

Se establece como fecha de entrega de la evaluación 3, el día lunes 27 de julio, antes de las 23:55 hrs.

Solo será necesario que entreguen el archivo 1 de los miembros del equipo de trabajo, indicando a través de un comentario multilínea los integrantes y respetando el siguiente formato:

```
1  /*
2      Evaluación 3 - Algoritmos y programación
3      Integrantes:
4          - Nombre alumno 1 - Rut alumno 1
5          - Nombre alumno 2 - Rut alumno 2
6      Sección: N° sección
7  */
```

En caso de que el trabajo sea realizado de forma individual, se deberá respetar el formato anterior, pero indicando solo al alumno que lo realizó.

El formato de entrega será un archivo fuente C (extensión **C**) que deberá ser subido al ambiente de aprendizaje antes de la fecha indicada anteriormente.

**Importante:** Bajo ningún punto se recibirán trabajos fuera de fecha o a través de otro medio que no sea el indicado anteriormente (AAI). Cualquier trabajo que no respete el formato indicado en la evaluación o fuera de fecha **no será revisado**.