



# Programación Concurrente y Distribuida (CC65)

*Trabajo Final 2023-2*

Secciones: Todas  
Profesores: Carlos Alberto Jara García

## Instrucciones

- El trabajo se desarrolla en grupos de mínimo 2 integrantes y máximo 3, se recomienda mantener los grupos formados en la primera mitad.
- Se debe entregar un repositorio github o similar, el cual debe ser subido en formato ZIP al aula virtual.
- Se debe entregar un informe de trabajo final, el cual debe ser subido al aula virtual.

## Consideraciones del Juego Ludo modificado

### Descripción del Juego:

El Ludo modificado es una versión ampliada y adaptada del popular juego de mesa Ludo. En esta versión, los jugadores compiten para llevar a sus personajes a través de un peligroso laberinto lleno de obstáculos y desafíos. Cada jugador tiene la tarea de guiar a sus personajes a través del laberinto y llegar a la meta antes que los demás.

### Reglas del Juego:

#### Tablero del Laberinto:

El tablero está dividido en casillas con caminos, obstáculos.

Cada jugador tiene cuatro personajes que comienzan en puntos de partida específicos.

#### Turnos y Movimientos:

Los jugadores se turnan para lanzar un dado y mover a sus personajes.

Los jugadores lanzan tres dados, dos dados normales (del 1 al 6) y uno con la operación (sumar o restar) para determinar cuántos pasos pueden avanzar o retroceder en su turno.

Los jugadores pueden mover un solo personaje por turno.

Los personajes deben avanzar exactamente la cantidad de pasos indicados por la operación de los dados (valor del primer dado y operador (+ -) con el valor del segundo dado).

**Obstáculos:**

El laberinto está lleno de obstáculos como paredes, trampas y criaturas que bloquean el paso de los personajes en varias casillas.

Si al personaje le toca avanzar hacia una casilla con obstáculo entonces el jugador pierde el turno y continua el siguiente jugador.

**Objetivo del juego:**

El objetivo es llevar a los cuatro personajes desde los puntos de partida hasta la meta en el menor número de turnos posible.

El primer jugador en llevar a todos sus personajes a la meta gana el juego.

**Modificaciones, uso de canales y nodos distribuidos:**

En esta versión modificada del juego, los jugadores y el tablero de juego están representados como entidades concurrentes separadas (nodos distribuidos) que se comunican a través puertos y mensajes json utilizando un mecanismo de sincronización de canales. Cada jugador tiene su propia bitácora con los datos de las IPs de todos los miembros de la red y utiliza un conjunto de puertos para comunicación con el tablero del juego y enviar movimientos y recibir actualizaciones del estado del juego.

**Comunicación:**

Cada jugador tiene un número de puertos de comunicación bidireccional con el tablero.

Los jugadores enviarán los resultados de los lanzamientos de dados y recibirán instrucciones para mover a sus personajes.

El juego deberá tener una interfaz Web para configurar el juego, iniciar y mostrar gráficamente el desarrollo del juego en tiempo real, deberá permitir mostrar los créditos y el resumen con los resultados al finalizar el juego.

**Lógica del Movimiento:**

Implementar la lógica para mover los personajes en el laberinto según las reglas del juego.

Manejar las interacciones con obstáculos.

### **Sincronización y Gestión de Turnos:**

Garantizar que los movimientos de los jugadores se manejen de manera sincronizada y que un jugador no pueda moverse fuera de su turno.

Implementar la gestión de turnos para alternar entre los jugadores.

### **Fin del Juego:**

Detectar cuando un jugador ha llevado a todos sus personajes a la meta y declarar a ese jugador como ganador.

Recuerda considerar la concurrencia y la sincronización al implementar las interacciones a través de los canales para garantizar que el juego funcione de manera justa y sin errores durante las operaciones simultáneas de los jugadores.

### **Objetivos del Trabajo**

- El objetivo del proyecto es simular el juego de niños Ludo modificado usando programación concurrente, canales y algoritmos distribuidos.
- La simulación debe correr concurrentemente usando algoritmo distribuido y manejar un grupo grande de personajes y jugadores como hosts donde la comunicación es a través de puertos y sincronización usando canales.
- La simulación debe mostrar el progreso del juego en tiempo real.
- Debe mostrar una vista Front End usando un framework de implementación páginas SPA para ingresar la configuración del juego con la cual se inicia el juego, una vista del progreso del juego en tiempo real, una vista de cuadro gráfico de estadísticas final del juego una vez finalizado y una vista de créditos del juego.

### **Alcance**

- La tarea debe ser implementada usando terminales virtuales para la red de nodos distribuidos.
- Es necesario implementar una interfaz gráfica usando páginas web SPA (Single Page Application).
- Debe hacer uso de un marco de trabajo ágil (SCRUM) y herramientas de gestión de versiones de código fuente. Utilizar GitHub como herramienta para estos fines.

## **Exposición**

La exposición forma parte de la nota. Si al momento de la exposición el profesor determina que el alumno no ha hecho parte o la totalidad del trabajo debido a que el alumno no supo responder correctamente a las preguntas realizadas el profesor podrá considerar descontar puntos en funcionalidades ya implementadas del trabajo. La frase “En esa parte me ayudaron” no será considerada como válida por lo que el alumno deberá realizar el trabajo de forma total.

## **Instrucciones para la entrega del trabajo**

La calificación del trabajo final es individual donde se medirá el rendimiento del alumno de acuerdo con la rúbrica que se entregará en clases.

El plazo es impostergable y por ningún motivo y/o circunstancia se recibirán trabajos fuera de esa fecha y hora.

El trabajo consiste en la elaboración de un informe, presentación, video y del código fuente.

## **Estructura del Informe**

Cada grupo debe entregar un informe detallando cada uno de las secciones que se muestran a continuación:

- Carátula
- Resumen
- Índice
- Objetivo del Estudiante (Student Outcome)
- Capítulo 1: Presentación
- Capítulo 2: Marco Teórico
- Capítulo 3: Implementación de solución
- Capítulo 4: Verificación de la solución
- Conclusiones
- Recomendaciones
- Glosario de términos
- Bibliografía (APA)
- Anexos (links video, github, otros)

## Rúbrica

	Sobresaliente	Satisfactorio	Deficiente
Planteamiento	3 puntos Usa diagramas para complementar la explicación de su planteamiento. Programación concurrente distribuida.	2 puntos Define procesos e Información.	0 puntos No elaborado
Implementación	5 puntos Funciona correctamente, utiliza nodos virtuales, puertos, para procesar los datos generados por el juego concurrentemente, usa canales y muestra resultados siguiendo las consideraciones del juego Ludo modificado.	3 puntos Elabora el código y ejecuta concurrentemente, usa puertos, canales, muestra resultados del juego, implementando las consideraciones del juego Ludo modificado.	0 puntos No elaborado
Interfaz	5 puntos La interfaz en modo gráfico configura el juego, inicia, muestra el avance del juego en tiempo real y muestra la estadística del juego finalizado usando gráficos (tortas, barras, etc) con los créditos del juego de manera clara y organizada en página SPA.	3 puntos Muestra el funcionamiento de la aplicación y resultados del juego entendible en página web.	0 puntos No elaborado
Informe	4 puntos Hace buen uso del medio escrito, cumple con la estructura, guarda coherencia y presenta conclusiones entendibles y orientadas a resultados. Anexa el resultado de la	3 puntos Sigue una estructura clara y contiene los elementos mínimos necesarios solicitados. Usa herramienta de versionado de código.	0 puntos No elaborado

	gestión del trabajo utilizando el marco SCRUM y el historial del versionamiento del código.		
Video	3 puntos Elabora un video de 6 minutos como máximo donde el grupo presenta el desarrollo y funcionamiento del juego, demostrando dominio del tema, la gestión del trabajo final usando el marco SCRUM.	2 puntos Cubre los puntos Solicitados y evidencia la gestión del trabajo final usando el marco SCRUM.	0 puntos No elaborar el video afecta la calificación en los demás ítems de la rúbrica.

Santiago de Surco, noviembre de 2023