

[AA1] Práctica - Bloque 1: TCP

El objetivo de esta práctica es:

- Implementar el juego sobre la topología P2P viendo sus ventajas e inconvenientes
- Programar sobre una librería de sockets a bajo nivel para implementar un juego por turnos
- Desvincular el código de redes de la lógica del juego

Paso 1: Funcionamiento del juego a implementar

En esta ocasión se ha escogido un juego que puede jugarse a través de una interfaz en modo texto: El juego de Virus.

Si no habéis jugado nunca a este juego, podéis revisar el siguiente enlace en el que se explican las reglas del juego y se hace un ejemplo de partida: <https://superjuegosdemesa.com/virus/>

Las reglas del juego son las siguientes:

<https://tranijsgames.com/wp-content/uploads/2017/02/VIRUS-RULES-esp.pdf>

Estos son los diferentes tipos de cartas que hay:



Se pueden categorizar en:

- Órganos: Estómago, Cerebro, Esqueleto y Corazón
- Medicinas: Tratamiento para cada uno de los órganos
- Virus: Virus también para cada uno de los órganos.
- Comodines:
 - o Existe un órgano que se puede utilizar como cualquiera de los cuatro órganos existentes en el juego
 - o Existe un tratamiento que se puede utilizar para curar o inmunizar a cualquiera de los órganos, pero también puede ser neutralizado con cualquiera de los virus.
 - o Existe un virus que se puede utilizar para infectar a cualquiera de los órganos (y también se puede curar con cualquiera de los tratamientos).
- Tratamientos: Son cartas que permiten hacer acciones especiales como las siguientes:
 - o Contagio
 - o Ladrón de órganos
 - o Trasplante
 - o Guante de látex
 - o Error médico

Lee el PDF de reglas para más información sobre la utilización de estas cartas.

Este es el número de cartas que hay de cada tipo en la baraja.

Órganos	1	5	5	5	5
Virus	1	4	4	4	4
Medicinas	4	4	4	4	4
Tratamientos	2	3	3	1	1

Preparación de la partida

- Se reparten todas las cartas entre todos los jugadores. Para nosotros las partidas serán de 2 a 4 jugadores.
- Cada jugador ve sus cartas y no sabe qué cartas tienen los demás.

Empieza el turno el jugador que tiene más órganos. En caso de empate, empieza el primer jugador que se conectó a la partida.

Qué ocurre en un turno

El jugador que tiene el turno puede elegir entre una de estas acciones:

- 1) Si tiene un órgano, ponerlo sobre la mesa.

- a) Si al tirar este órgano, este jugador completa los 4 órganos necesarios y ninguno de ellos está infectado, **gana este jugador y se acaba la partida.**
- b) En caso contrario, roba una carta y pasa el turno al siguiente jugador
- 2) Puede infectar un órgano de otro jugador con el virus correspondiente. Roba carta y pasa el turno al siguiente jugador.
- 3) Puede vacunar o inmunizar uno de los órganos que previamente haya colocado sobre la mesa. Roba carta y pasa el turno al siguiente jugador.
- 4) Puede descartarse de una a tres cartas y robar el mismo número de cartas. Pasa el turno al siguiente jugador.
- 5) Puede echar una carta de tratamiento con los efectos que esa carta conlleve. Roba carta y pasa el turno al siguiente jugador.

Cuándo acaba la partida

- Cuando uno de los jugadores tiene los 4 órganos sobre la mesa y ninguno de ellos está infectado.
- La partida también finalizaría si quedara sólo un jugador en la partida porque el resto se hubiera desconectado.

Qué pasa si se han acabado las cartas que puedo robar

- Se barajan las cartas que se han descartado y pasar a ser cartas para robar.

Qué pasa si un jugador se desconecta

- Tanto las cartas que hay sobre la mesa como las que tiene en mano, vuelven a la baraja de cartas.
- Se debe tener en cuenta a la hora de asignar los turnos.
- Se deben liberar los recursos asociados a esta conexión.

Paso 2: Protocolo de comunicación

Indica qué mensajes deben intercambiarse los diferentes nodos de juego.

Especifica cuáles de estos mensajes se intercambian con el BSS en una tabla separada.

Sigue la siguiente estructura:

Estructura mensaje	Acción asociada

Paso 3: Implementación del juego

Establecimiento de conexión

Tenemos un BSS que cuando uno se conecta le dice si quiere iniciar partida o unirse a una.

Si quiere crear, se anota en una lista de partidas en espera. A la hora de crear una partida se debe poder especificar: nombre de la partida, número de jugadores (entre 2 y 4) y si la partida tiene *password* y cuál, si procede.

Si quiere unirse, se le muestran las partidas que están en curso y debe escoger una.

Cuando una partida se rellena con el número de jugadores que corresponde, sale del listado de partidas y empiezan su partida de forma separada del BSS.

Repartición de las cartas

No debe intervenir para nada el servidor BSS. Debe hacerse de forma independiente desde cada peer utilizando la misma *seed*.

Implementación de la lógica del juego

En base a tu protocolo de comunicación, implementa la funcionalidad del juego.

Deben repartirse correctamente las cartas

- Deben asignarse los turnos
- Desde cada nodo se debe recibir *feedback* de lo que está pasando en la partida, aunque no se tenga el turno: A quién le toca, qué carta se ha lanzado y a quién, si alguien se ha descartado y de cuántas cartas, si ha ganado alguien, si se ha desconectado un jugador, etc.
- La partida debe finalizar según los criterios indicados.

Organización del código

- Debe existir la funcionalidad del nodo y la del servidor por separado
- Programar la parte de redes separada de la lógica del juego de forma que, si se quisiera cambiar la librería de red que hay por debajo, pudiera hacerse sin problemas.
- Deben controlarse los errores de comunicación por red.

Entregable de la práctica → Protocolo

1. Documento Excel o similar en el que se detalle el protocolo utilizado para la topología P2P. El protocolo debe contemplar la creación/unión de/a partidas.

Entregable de la práctica → Código

1. Nos aseguramos de que la solución compila.
2. Eliminamos la carpeta oculta .vs
3. Hacemos un zip o 7zip de la solución

Lo subimos todo al Classlife en este formato:

AA1_Practica_Nombre1Apellido1Nombre2Apellido2.zip

Rúbrica

<https://docs.google.com/spreadsheets/d/1HV-uU7ImzckZHnXohWyllxHrql4GhGNkocNzAA6gb3w/edit?usp=sharing>