

INTRODUCCIÓN

Desarrollar un programa que permita la comunicación hasta un máximo de 6 computadores, en donde les permita a estos computadores jugar el juego de cartas denominado **Virus!**.

OBJETIVOS

1. Aplicar los conocimientos adquiridos durante el curso, acerca del modelo de capas y sus protocolos, y la comunicación entre computadores en una red.
2. Solucionar, mediante software, los desafíos inherentes que conllevan el desarrollo de un sistema en red.

DESCRIPCIÓN

¿QUÉ ES EL JUEGO DE CARTAS VIRUS!?

Virus! es un juego de cartas desarrollado por Domingo Cabrero, Carlos López y Santi Santisteban y distribuido por ©Tranjis Games. Este se juega bajo la siguiente historia, disponible en [1].

“En el hospital Nuestra Señora de Tranjis, saltan las alarmas cuando los novatos del laboratorio se dan cuenta demasiado tarde de que los contenedores de muestras no estaban vacíos como pensaban. En su interior contienen brotes de virus experimentales que ahora campan a sus anchas por todo el centro y sólo tú puedes detenerlos.

Tu misión consiste en enfrentarte con arrojo a la pandemia y competir por ser el primero en erradicar los virus logrando aislar un cuerpo sano para evitar la propagación de las terribles enfermedades. Éticos o no, todos los medios a tu alcance valen para ganar. El vencedor será colmado de reconocimientos y los perdedores arrojados a las enfurecidas autoridades sanitarias.



Figura 1. Juego de carta Virus!. [2]

El sistema deberá ser desarrollado utilizando las siguientes características fundamentales:

1. Contar con un mazo de 68 cartas divididas en 4 tipos:
 1. **Órganos:** Corresponden a cuatro órganos sanos de distinto color; corazón → rojo, estómago → verde, cerebro → azul, hueso → amarillo y órgano multicolor → actúa como comodín para ayudar a completar un cuerpo y cuenta como órgano a todos sus efectos.
 2. **Virus:** Son cartas utilizadas para infectar y extirpar los órganos y destruir medicinas de los rivales. Solo tienen efecto sobre órganos y medicinas que correspondan con el mismo color, también existe el virus multicolor, utilizado como virus comodín, puede infectar un órgano o destruir una medicina de cualquier color.
 3. **Medicinas:** Son cartas que sirven para proteger los órganos de las amenazas de virus de los demás jugadores. Estas pueden curar (destruye un virus), vacunar un órgano e inmunizar el órgano. Asimismo, existe el comodín de medicina multicolor, utilizadas para curar virus y vacunar órganos de cualquier color.
 4. **Tratamientos:** Este tipo de cartas pueden alterar el juego, permitiendo al jugador que las usa evitar que los rivales completen su cuerpo. Existen 5 tipos de Tratamientos: Trasplante, Ladrón de órganos, Contagio, Guante de látex y Error médico.
2. Cada división de tipos de carta, se dividen de la siguiente manera
 1. 21 cartas de órganos (5 corazones, 5 estómagos, 5 cerebros, 5 huesos y 1 órgano comodín)
 2. 17 cartas de virus (4 rojos, 4 verdes, 4 azules, 4 amarillos y 1 comodín)
 3. 20 cartas de medicinas (4 rojos, 4 verdes, 4 azules, 4 amarillos y 4 comodín)
 4. 10 cartas de Tratamientos (2 Trasplante, 3 Ladrón de órganos, 3 Contagio, 1 Guante de látex y 1 Error médico).

MODO DE JUEGO

Antes de comenzar

1. Se baraja el mazo de cartas y se reparten 3 cartas a cada jugador. Hasta un máximo de 6 jugadores.
2. Se coloca el mazo restante boca abajo para que todos los jugadores puedan coger cartas para jugar.
3. A la par del mazo se colocará la pila de descarte (cartas que ya se hayan jugado).
4. Cuando el mazo se haya agotado, se tomará las cartas de la pila de descarte, se barajará y ahora formará el nuevo mazo.

DESARROLLO DEL JUEGO

1. Cada jugador debe tener siempre al comienzo de su turno 3 cartas en mano y tan sólo podrá realizar una acción por turno.
2. Cada jugador podrá usar una de las 3 cartas que tenga en su mano.
3. Después de haber realizado alguna acción, tomará cartas del mazo para volver a tener su mano completa.
4. En cada turno del jugador se juega cualquiera de los distintos tipos de carta que tenga en su mano, colocándose sobre la mesa, si:
 - 4.1. Es un **Órgano**, la carta se colocada enfrente del jugador hasta completar 4 órganos, pero no se puede tener

en su cuerpo dos órganos iguales en ningún momento. **Nota:** Un cuerpo podrá tener cinco órganos distintos si uno de ellos es el multicolor (comodín), y se puede ganar la partida si cuatro de ellos están sanos.

4.2. Es una **Medicina**, la carta se coloca enfrente del jugador sobre un órgano, aquí ocurren 3 situaciones:

- 4.2.1. Curar → si sobre el órgano hay una carta de virus del mismo color a la medicina, entonces el virus se destruye y ambas cartas se envían a la pila de descarte.
- 4.2.2. Vacunar → si el órgano del mismo color a la medicina no posee ningún virus sobre él, el órgano se protege, ahora los demás jugadores requerirán 2 virus para poder infectar el órgano.
- 4.2.3. Inmunizar → si ya el órgano cuenta con una medicina, esta segunda medicina logrará proteger para siempre contra el ataque de cualquier virus y no podrá ser destruido ni afectado por cartas de tratamiento. Cuando el órgano se inmuniza las cartas de medicina se giran 90 grados sobre el órgano para indicar que está inmune.

4.3. Es un **Virus**, la carta se coloca sobre los cuerpos de los demás jugadores, se dan 3 situaciones:

- 4.3.1. Infectar → si se encuentra un órgano libre de virus y que corresponde con el color de la carta de virus a utilizar, este órgano se infectará.
- 4.3.2. Extirpar → si un segundo virus es colocado sobre un órgano ya infectado, este órgano será destruido y las tres cartas (el órgano y los 2 virus) serán enviadas a la pila de descarte.
- 4.3.3. Destruir vacuna → si sobre un órgano se encuentra una carta de medicina y se le aplica un virus del mismo color, ambas cartas (la medicina y el virus) serán enviadas a la pila de descarte.

4.4. Es un **Tratamiento**, la carta utilizada podrá desempeñar una acción de ataque hacia los demás jugadores, esta carta podrá desempeñar cualquiera de las siguientes funciones:

- 4.4.1. Trasplante: Permite intercambiar un órgano por otro entre 2 jugadores cualesquiera. No importa si el color de estos órganos es diferente, ni si están sanos, infectados o vacunados. Sencillamente el jugador cambia el órgano escogido por otro, siempre y cuando ninguno de los dos jugadores tenga dos órganos del mismo color ni éstos estén inmunizados.
- 4.4.2. Ladrón de órganos: Permite robar un órgano de otro jugador y añadirlo a su propio cuerpo. Se puede robar órganos sanos, vacunados o infectados, pero no inmunes. No se pueden tener dos órganos del mismo color.
- 4.4.3. Contagio: Permite trasladar tantos virus como se pueda de sus órganos infectados a los órganos de los demás jugadores. No se puede utilizar el contagio sobre órganos vacunados o infectados, sólo se podrá contagiar órganos libres.
- 4.4.4. Guante de látex: Permite que todos los jugadores, excepto el que utiliza el guante, descarten su mano. Al comienzo de su siguiente turno, al no tener cartas, estos jugadores tan solo podrán robar una nueva mano, perdiendo así un turno de juego.
- 4.4.5. Error médico: Permite intercambiar todo su cuerpo por el de otro jugador, incluyendo órganos, virus y vacunas. No importa el número de cartas que cada uno tenga en mesa. Cuando se utiliza esta carta, los órganos inmunizados también son intercambiados.

5. Si el jugador en turno no puede realizar ninguna de las acciones anteriores o simplemente desea descartar algunas o todas las cartas de su mano, entonces descartará en la pila de descarte tantas cartas desee (1, 2 o las 3 cartas), luego deberá sustituirlas por la misma cantidad de cartas tomadas del mazo y finalmente pasar de turno.

FINALIZACIÓN DEL JUEGO

1. El primer jugador que tenga delante de él, en su cuerpo 4 órganos sanos será automáticamente el ganador.

ASPECTOS TÉCNICOS

1. El Proyecto puede ser desarrollado utilizando los siguientes lenguajes de programación: C++, C#, Java o Python 3. Aplicando cualquier tipo de interfaz, ya sea en aplicación de escritorio o web.
2. El Proyecto debe implementarse utilizando el nivel de abstracción más bajo para el paso de mensajes en red, denominado Socket, además es necesario utilizar hilos de ejecución (threads).
3. Se debe aplicar el modelo Cliente/Servidor, en donde se tendrá un equipo dedicado con funcionalidades de servidor, en éste se gestionará el funcionamiento del juego y almacenamiento de los datos. No se permite utilizar un servidor ya construido, como por ejemplo un Apache, Nginx, Microsoft IIS, etc. El cliente se conectará al servidor y solicitará ingresar al juego.
4. El sistema debe estar funcionando en una Red Virtual Privada (Hamachi), en donde los clientes se conectarán a un dispositivo conectado en la red con rol de servidor y es el que gestionará el funcionamiento del programa y en donde los clientes consumirán los datos. Al momento de la defensa deberán existir mínimo 3 equipos reales (jugadores) conectados en la red que permitan realizar una partida. En la siguiente figura se puede observar el ejemplo de la red virtual utilizando el modelo Cliente/Servidor.

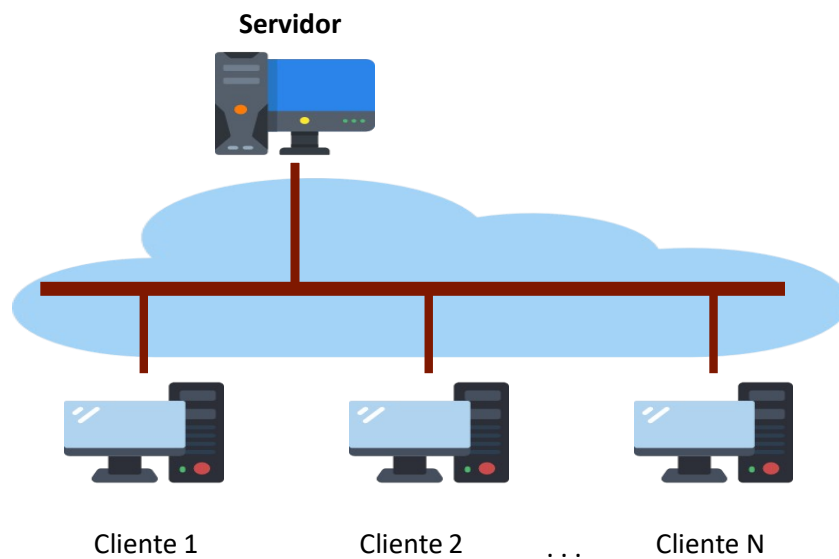


Figura 2. Arquitectura Cliente/Servidor

5. Debido a que todas las ilustraciones del juego de cartas son propiedad del autor y cedidas a TRANJIS GAMES S.L. y por lo tanto, queda prohibida su reproducción sin el expreso consentimiento del artista, se deberán realizar cada una de las ilustraciones requeridas para el juego. Es decir, cada grupo deberá utilizar su creatividad para desarrollar las ilustraciones de cada una de las cartas que permitan cumplir satisfactoriamente con cada una de las funcionalidades a las que se les indiquen.

ASPECTOS ADMINISTRATIVOS

- Lea y comprenda cuidadosamente lo que se le solicite.
- El Proyecto será realizado en parejas o grupos de 3 personas.
- El día de entrega, el proyecto será defendido de manera individual. El demostrar poco conocimiento sobre el desarrollo del proyecto o si se determina que alguno de los miembros del grupo trabajó menos, se penalizará a todos miembros con un 20% de la nota final obtenida.
- No se permite entregar el Proyecto en partes.
- Si se comprueba que existen dos o más Proyectos similares se procederá a colocar nota cero a todos los proyectos involucrados.
- Subir los archivos al Aula Virtual, todos los archivos que conforman el proyecto deben estar dentro de una carpeta comprimida en formato *.zip y cumpliendo con el formato establecido en el Programa de Curso, disponible en el Aula Virtual.
- Al momento de probar el Proyecto debe estar funcionando en una Red Virtual Privada, utilizando equipos personales (como mínimo se deben contar con 3 equipos conectados).
- Fecha de Entrega
 - Entrega del enunciado: 18 de abril del 2020.
 - Revisión y defensa: 17 de junio del 2020.

CRITERIOS DE EVALUACIÓN

Como directriz de la cátedra se establece como obligatoria la defensa del proyecto, por parte de todos los miembros del grupo, debido a los cambios sufridos en el curso por la virtualización de éste, se estará informado la manera en que se procederá esta defensa, pero los criterios de evaluación correspondiente para esta entrega se muestran a continuación (para un total de 100%).

1. Eficacia (cumple con el funcionamiento solicitado) (70%).
2. Tratamiento de errores y validación (10%)
3. Eficiencia y estructura solicitada (10%)
4. Facilidad de Uso y Creatividad/Originalidad (GUI) (8%)
5. Entrega de Fuentes (2%)

En el Anexo 1 se indica la especificación de los criterios a evaluar para la revisión del proyecto.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Tranjis Games. *Virus! El juego de cartas más contagioso*. 2020. Disponible en <https://tranjisgames.com/wp-content/uploads/2017/02/VIRUS-RULES-esp.pdf>.
- [2] Tranjis Games. *Imagen de la caja del juego de cartas*. 2020. Disponible en <https://tranjisgames.com/tienda/juego-de-mesa-virus/>.

ANEXO 1.

1. Eficacia (Cumple el funcionamiento solicitado) (70%).

	Deficiente		Regular		Satisfactorio	
RUBRO	0	1	2	3	4	5
1. General (20%)						
1.1. Se cumplen las reglas del juego a cabalidad (10 %)						
1.2. Diseño de las ilustraciones acordes a las funcionalidades que desempeñan (3 %)						
1.3. El sistema funciona sobre una red VPN (2 %)						
1.4. El sistema determina el ganador del juego (5 %)						
2. Servidor (26%)						
2.1. El programa le permite a los clientes (jugadores) conectarse (5 %)						
2.2. El programa realiza la gestión del juego, permitiendo que los clientes puedan jugar una partida (10 %)						
2.3. El programa permite recibir peticiones de los clientes y retornar los resultados esperados (10 %)						
2.4. El programa permite observar a los clientes que se encuentran conectados (1 %)						
3. Cliente (24%)						
3.1. El programa permite conectarse al servidor (4 %)						
3.2. El programa muestra la información enviada por el servidor y muestra el resultado esperado en pantalla (10 %)						
3.3. El programa muestra a los jugadores conectados y sus correspondientes cartas jugadas (10%)						

2. Eficiencia y estructura solicitada (25%)

Rubro	Deficiente 0 – 1	Regular 2 – 3	Satisfactorio 4 – 5
Separación lógica de capas	Se desarrolló el programa sin realizar ninguna separación lógica de capas	Se utilizan capas, pero no hay una división clara de lógicas	Las capas utilizadas están claramente separadas según su lógica.
Tratamiento de errores y validación	El programa se detiene continuamente porque no tiene control de errores o este es mínimo	El programa tiene control de muchos errores a nivel de código, pero se ve interrumpido debido a la manipulación del usuario. Presenta errores de lógica que desembocan en comportamientos inesperados	El programa tiene el control interno de todos los errores predecibles, e informa al usuario de los errores que surjan por la manipulación del programa

3. Entrega de Fuentes y Creatividad/Originalidad (5%)

Rubro	Deficiente 0 – 1	Regular 2 – 3	Satisfactorio 4 – 5
Facilidad de uso	El sistema es difícil o imposible de utilizar porque el usuario necesita tener conocimiento previo para usarlo, o bien la app devuelve errores difíciles de resolver	El usuario tiene que investigar cómo utilizarlo, las opciones no están a mano y el sistema responde de manera diferente a las acciones que el usuario espera	El usuario puede realizar todas las funciones que desea sin ser inducido a error y puede encontrar fácilmente lo que necesita
Creatividad y Originalidad	En el proyecto no se demuestra innovación		Capacidad de generar nuevas ideas o conceptos que produjeron soluciones originales
Fuentes	El Proyecto no permite ser ejecutado		El Proyecto es completamente funcional