

## [AA4] - Práctica - Bloque 2: UDP

### Ítem 1: Diseño de un juego multiplayer online

A partir de las temáticas que se ofrecen en el PDF PosiblesJuegosAA4, diseño un juego para al menos 2 jugadores, que tenga sentido jugarlo online.

Dado que TCP funciona mejor en juegos que no utilizan tiempo real, el juego que diseñes no debe ser necesario jugarlo en tiempo real o el tiempo real no debe ser determinante.

- La partida debe empezar una vez llegan todos los jugadores.
- Las condiciones de finalización del juego dependerán de las reglas que defináis.

**El juego debe tener interfaz gráfica, aunque sea muy sencilla.**

Como os basáis en uno de los juegos propuestos, para esta entrega vais a implementar un juego que utiliza movimiento en tiempo real. Con el objetivo de optimizar la comunicación, en este caso, vamos a utilizar **sockets UDP** para implementar la parte de red.

Otro punto a tener en cuenta, es que únicamente vamos a implementar este juego en topología cliente/servidor.

### Ítem 2: Prepara la base de datos

Para cada uno de los juegos se indicaba una base de datos a diseñar y preparar con datos de prueba. En este primer ítem se debe dejar preparada la base de datos, lista para poder empezar a trabajar con ella.

### Ítem 3: Define el protocolo de comunicación

Una vez diseñado el juego y sabiendo las fases por las que pasa desde que llegan los jugadores a la partida hasta que se dan las condiciones de victoria/derrota, debes diseñar el protocolo de comunicación que se utilizará.

Ten en cuenta cada una de estas fases para diseñar el protocolo.

#### Fase 1: Establecimiento de la conexión y condiciones iniciales

Esto se refiere a cuando los jugadores se conectan al servidor y éste les indica sus condiciones iniciales de juego y las de los demás jugadores, si es el caso.

Explica si en esta fase hay algún mensaje que sea crítico y cómo aseguras que llega a su destino.

Es posible que en tu juego los jugadores puedan conectarse y desconectarse en cualquier momento sin que eso afecte al desarrollo del juego. Igualmente, trataremos esta fase por separado a nivel de protocolo.

Explica también en este punto cómo se trata el caso de que un jugador se desconecte a mitad de partida.

Esta fase de establecimiento de conexión debe incluir una validación de usuario contra base de datos. De esta base de datos se debe extraer al menos un valor/criterio/parámetro que sirva para implementar *matchmaking* (ítem 5).

## Fase 2: Desarrollo del juego

Indica qué mensajes son necesarios para el desarrollo del juego y cuáles son críticos. Al menos uno de los mensajes que se intercambie durante el desarrollo del juego debe ser crítico y otro mensaje no debe serlo. Cada uno debe ser tratado como corresponda.

Indica qué mensajes son los que sirven para evitar la pérdida de paquetes.

Indica qué mensajes son los que sirven para construir el Client Prediction.

Indica qué mensajes son los que sirven para construir el Client Interpolation.

## Fase 3: Finalización

Indica qué mensajes intercambian cliente y servidor para marcar la finalización del juego y el ganador/perdedor.

## Listado de mensajes

Estructura mensaje	Sentido	Acción asociada

Recuerda añadir al código lo necesario para poder simular la pérdida de paquetes y comprobar que tu protocolo se recupera correctamente.

## Ítem 4: Optimización del protocolo de comunicación

Para cada uno de los mensajes definidos en tu protocolo, indica qué optimizaciones has realizado a nivel de tipo y a nivel de bit. Dejo un ejemplo:

Estructura mensaje	Tipos	Número de bits	Explicación
HELLO_<Nick>	Int_Int_string	3_3_88888888	8 tipos de paquetes diferentes (3 bits) Máxima longitud del Nick: 8 caracteres (3 bits) Cada uno de los caracteres del Nick (8 bits)

## Ítem 5: Client Prediction & Interpolation

Implementa las estructuras necesarias para evitar el lag en el movimiento del propio jugador y en el movimiento de los demás jugadores.

## Ítem 6: Matchmaking & Bases de datos

En este punto, se debe reorganizar (si no lo has hecho ya) el código del servidor de para que admita más de una partida a la vez.

En la práctica de TCP (AA3), las partidas las creaba cada jugador y también era el mismo jugador el que decidía cuándo unirse a una partida.

En esta práctica, el jugador sólo decidirá cuándo quiere jugar. Cuándo lo indique, será el propio servidor el que se buscará un contrincante apropiado a la característica por la que se está realizando *matchmaking*.

Al acabar una partida, al jugador se le debe dar la posibilidad de jugar otra.

### Importante (próximamente se publicará la rúbrica "oficial")

A la hora de evaluar se considerarán los siguientes puntos:

1. Que el protocolo de comunicación diseñado cubra todas las situaciones del juego
2. Que el juego escogido ofrezca una variedad de situaciones lo suficientemente relevante
3. Que el protocolo de comunicación esté correctamente implementado
4. La implementación de la comunicación por red: Entre jugadores y servidor
5. Que estén controlados los errores de comunicación: Sobre todo a la hora de conectarse y desconectarse (acabar partida)
6. La solución a la pérdida de paquetes
7. Indica desde dónde se puede cambiar la probabilidad de pérdida de paquete
8. La optimización a nivel de bit
9. La solución propuesta en Client Prediction
10. La solución propuesta en Client Interpolation
11. La implementación del acceso a base de datos.
12. La implementación de la base de datos.
13. La implementación del *matchmaking*.

También se valorará:

1. El acabado final

### Entregable de la práctica → Código

1. Nos aseguramos de que el proyecto no utiliza paths absolutos para linkar las librerías.
2. Nos aseguramos de eliminar la carpeta oculta .vs.
3. Hacemos un zip de la solución
4. Lo subimos al classlife en este formato:

PracticaUDP\_Nombre1Apellido1Nombre2Apellido2.zip

### Entregable de la práctica → Documentación

1. Protocolo de comunicación según lo que se explica en el ítem 3 del enunciado.
2. Explica la optimización que has aplicado en el protocolo según se indica en el ítem 4.
3. Explica/Dibuja las estructuras de datos que intervienen en la recuperación de paquetes perdidos.
4. Explica/Dibuja las estructuras de datos que intervienen en el Client Prediction
5. Explica/Dibuja las estructuras de datos que intervienen en el Client Interpolation
6. Dificultades que has encontrado y conclusiones.