**Indicaciones de entrega**

Una vez realizada la tarea elaborarás un único fichero **.zip** donde figuren las respuestas correspondientes. El envío se realizará a través de la plataforma de la forma establecida para ello, y el archivo se nombrará siguiendo las siguientes pautas:

**apellido1\_apellido2\_nombre\_SIGxx\_Tarea**

Asegúrate que el nombre no contenga la letra ñ, tildes ni caracteres especiales extraños. Así por ejemplo la alumna **Begoña Sánchez Mañas para la primera unidad del MP de DAW**, debería nombrar esta tarea como...

**sanchez\_manas\_begona\_PSP\_Tarea01**

**Cada ejercicio ha de desarrollarse en un documento distinto, siguiendo los nombres de ej1, ej2,..**

**Ejercicios**

1. **(3 pts) Crea dos clases (cliente y servidor) para que a través de UDP el cliente envíe un comando al servidor y éste le responda una determinada acción.**

Los comandos serán:

* getdate: el servidor enviará la fecha actual
* gethour: el servidor le enviará la hora actual
* getmayus(“xxx”): el servidor retornará la cadena xxx en mayúsculas

Tener controlados los puertos de entrada y salida para la correcta comunicación

Cuidado el manejo de Strings y formatos

Ejemplo Cliente:

Introduce comando:

getdate

Creando paquete… ok!

Enviando paquete a “ip” por puerto”port”… ok!

Esperando respuesta…

Respuesta: fecha 28/01/2022

Ejemplo Servidor:

Iniciando servidor…

Ready!

Puerto xxx en escucha

Recibido mensaje de “ip”: “mensaje”

Enviando respuesta: “respuesta”

1. **(1.5 pts)** Crear una clase cliente que se una al **grupo multicast** del servidor anterior. ejecutar varias instancias de cliente y verificar su funcionamiento
2. **(3 pts) Crear un chat UDP simple** en el que un cliente se conecte con un nombre de usuario y el servidor recibirá los mensajes enviados por cada cliente y los enviará por el grupo Multicast para que lo reciban todos.

Puedes hacerlo a través de la consola de comandos utilizando colores o bien implementar una interfaz gráfica.

**Ejemplo cliente:**

***Introduce nombre de usuario:***

***pepe***

***Bienvenido pepe, ya puedes comenzar a chatear.***

***jose: hola a todos***

***Juan: ¿que tal?***

***buenos días***

***yo: buenos días****.*

1. **(2.5 pts)** Realiza mediante una interfaz gráfica un cliente que permita introducir una serie de datos de una persona que serán enviados a un servidor. El servidor debe mostrar los datos recibidos por cada cliente atendidos mediante un Hilo. Deberás crear la clase persona serializable con los atributos necesarios.

