**PSP03 TE01**

**ENUNCIADO**

Juego Adivina Número

La presente tarea consiste en adivinar un número aleatorio creado por el servidor y que pueden adivinar entre los clientes conectados a esa jugada.

SERVIDOR

* El servidor envía un mensaje al cliente para que el usuario adivine un número entre el 1 y el 100.
* El servidor aceptará diferentes conexiones y por cada una de ellas creará un hilo.
* Revisa si el cliente ha acertado el número aleatorio creado al inicio del juego.
* Revisa si el juego ha terminado. Si el juego ha terminado, notifica a los clientes de que el juego ha finalizado y han adivinado el número creado de forma aleatoria mediante un random desde el servidor.
* En caso de ocurrir un error en el programa se notifica al cliente.

CLIENTE

* Comprueba si no ha terminado el juego o se ha dado un error.
* Recoge la respuesta del servidor y la muestra.
* En el caso de que haya acertado mostrará: Has ganado!! El juego ha terminado.
* En caso de que otro jugador haya ganado mostrará: El juego ha terminado, han adivinado el número. Y el identificador del cliente que ha ganado.
* Cada cliente tendrá un identificador, puede estar asociado al socket de cada cliente o puede ser un número aleatorio.
* Al finalizar la partida el cliente mostrará el número de intentos que ha realizado.

En esta tarea debes realizar una única solución que contenga dos proyectos. El primer proyecto será el cliente y el segundo el servidor. Ambos proyectos deben comunicarse haciendo uso de sockets TCP. Se pueden utilizar sockets síncronos o asíncronos, pero el servidor debe ser capaz de dar servicio a un máximo de 10 clientes. Los clientes podrán trabajar de forma simultánea y debes tener cuidado de que no haya bloqueos.

Vídeo del Ejercicio

**PSP03 Tarea Evaluativa Numero Aleatorio**

[](https://youtu.be/YHdse12EldE)

Enviar una Foto

Hacer una aplicación cliente-servidor en el que el cliente le solicite una imagen al servidor mediante un menú y éste le provea de dicho fichero. La aplicación deberá crearse con las clases socket y admitirá como máximo 5 clientes simultáneos.

CLIENTE

* El usuario se conectará a la consola mostrada por la parte cliente en el que aparecerá un menú.
  1. 1.- FotoMonte
  2. 2.- FotoPlaya
  3. 3.- FotoCiudad
  4. 4.- Salir
* El cliente deberá seleccionar una foto y enviar la petición al servidor.
* El cliente esperará la recepción de la foto. Y cuando la descargue la guardará en un directorio local.
* Una vez guardado en un directorio local, se muestra el path/ruta al usuario para que este pueda visualizar la imagen descargada.
* Saldremos del menú si, pulsamos la opción de salir, de lo contrario el menú se mostrará indefinidamente.

SERVIDOR

* El servidor recibe la petición del programa cliente.
* Recoge la imagen guardadas de antemano en un directorio local y se la envía por red al cliente.

Este ejercicio se puede realizar con entorno gráfico si se desea, pero no será una condición.

[FotoMonte](https://ikastaroak.birt.eus/pluginfile.php/245383/mod_assign/intro/FotoMonte.jpg), [FotoPlaya](https://ikastaroak.birt.eus/pluginfile.php/245383/mod_assign/intro/FotoPlaya.jpg) y [FotoCiudad](https://ikastaroak.birt.eus/pluginfile.php/245383/mod_assign/intro/FotoCiudad.jpg)

Documentación

Como tercer ejercicio se elaborará documentación que contendrá la siguiente información de ambos programas.

* Funcionalidad Ejercicio 1 y Ejercicio2.
* Aspectos relevantes del desarrollo del código.
* Batería de pruebas empleada.
* Pequeña autoevaluación.

**RECURSOS**

* Ordenador personal
* Entorno de desarrollo Visual Studio 2022
* .Net Core 2.1 o superior
* Apuntes y repositorio del módulo

**CONSEJOS Y RECOMENDACIONES**

Juego Adivina Número

Aunque no sea obligatorio la creación de los siguientes objetos puede ayudar al desarrollo de la tarea:

Objeto Compartido con todos los hilos para el análisis de los datos

* int número: número a adivinar.
* int o similar ganador: contendrá el identificador del ganador
* Int o similar participantes: Array que contiene todos los identificadores de todos los participantes
* int numParticipantes: El número de participantes que están participando.
* Cualquiera que os ayude en la ejecución del programa

Objeto con datos de cada jugador:

* int o similar identificador: contendrá el identificador del ganador
* int intentos: intentos que lleva el jugador.
* Cualquiera que os ayude en la ejecución del programa

Vídeo de la unidad

**Recordatorio - Lock**

[](https://youtu.be/_6fKquPun0A)

Indicaciones de entrega

Una vez realizada la tarea incluiras en una carpeta la solución que contiene los dos proyectos. 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 tercera unidad del MP de PSP,** debería nombrar esta tarea como...

**sanchez\_manas\_begona\_PSP03\_Tarea**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Muy bueno (10pt)** | **Bueno (8-9pt)** | **Satisfactorio (5-7pt)** | **Debe mejorar (4pt)** |
| **Ejercicio1 (Adivina Número)** | El servidor acepta varias conexiones cliente  Entran en juego todos los clientes y los datos que se manejan son consistentes.  Controla el final del juego. Cuando finaliza un usuario, avisa al resto de final de juego.  Gestiona los usuarios. Identificación de usuarios.  Conexión cliente-servidor | El servidor acepta varias conexiones cliente  Controla el final del juego  Gestiona los usuarios | El servidor gestiona el juego, pero no acepta varios clientes. | El servidor no acepta varios clientes, ni gestiona el juego correctamente |
| **Ejercicio2 (Descarga Imagen)** | Solicitud correcta al servidor de imagen  Descarga correcta de imagen  Conexión cliente-servidor mediante socket. | Conexión cliente-servidor correcta pero algún fallo de gestión. | Problemas con la conexión cliente-servidor  Problemas con la conexión del stream | No funciona el programa |
| **Documentación** | Se muestra funcionalidad Ejercicio 1 y Ejercicio2.  Se muestra aspectos relevantes del desarrollo del código.  Se muestra la batería de pruebas empleada.  Se muestra la autoevaluación. | No se desarrollan todos los puntos solicitados, pero sí los más relevantes de ambos ejercicios. | Se muestran los aspectos más relevantes de un ejercicio. | No existe documentación o apenas hay información relevante de ambos ejercicios. |

**CRITERIOS DE CALIFICACIÓN. TOTAL 10 PUNTOS.**

|  |
| --- |
|  |
| Observaciones: |

* Total 10 puntos.
  + Ejercicio 1. 5 puntos.
  + Ejercicio 2. 3 puntos.
  + Ejercicio 3 (Documentación). 2 puntos