# Práctica 1. Introducción a la programación de aplicaciones con las primitivas Socket en C para TCP.

TITULACIÓN: Grado en Ingeniería telemática (14512011)

TITULACIÓN: Doble Grado Ing. de tecnologías de la telecomunicación e Ing. telemática (15212023)

CENTRO: ESCUELA POLITÉCNICA SUPERIOR (LINARES)

CURSO ACADÉMICO: 2024-2025

## Índice

## Descripción

En esta práctica se introducirá en el estudio de las primitivas socket para la programación de clientes y servidores sobre el protocolo TCP en una red IP. Para ello se mostrará el funcionamiento de un cliente y un servidor sencillos sobre TCP cuyos códigos estarán disponibles en la plataforma de docencia virtual de la Universidad de Jaén. Para profundizar en el conocimiento sobre este tipo de aplicaciones, al código entregado se le deberá hacer modificaciones para adaptarlo a un nuevo protocolo de aplicación.

La práctica se estructurará en una serie de tareas distribuidas entre las sesiones de laboratorio establecidas para la práctica.

## Tarea 1. Introducción a la plataforma y programas Sockets

Esta tarea comprende las acciones iniciales para poder comenzar con el estudio, diseño e implementación de clientes y servidores TCP en lenguaje C con Microsoft Visual Studio.

1. Introducción a Microsoft Visual Studio.
2. Repaso y refresco de conocimientos del lenguaje C.
3. Crear dos proyectos en Microsoft Visual Studio:
    a. Para esto se deberán **abrir dos instancias de Microsoft Visual Studio** y crear cada proyecto en una instancia diferente.
    b. Elegir “Crear un proyecto” (también desde el menú Archivo – Nuevo Proyecto). Véase la Figura 1.
    c. Crear un proyecto vacío de C++ (ver Figura 2).
    d. Elija una carpeta destino y ponga a cada una un nombre fácilmente reconocible como **práctica1_cliente** y **práctica1_servidor**.

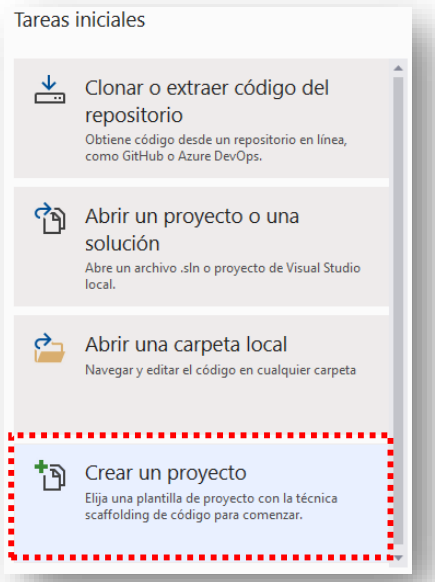


*Figura 1. Cuadro de creación de proyecto.*

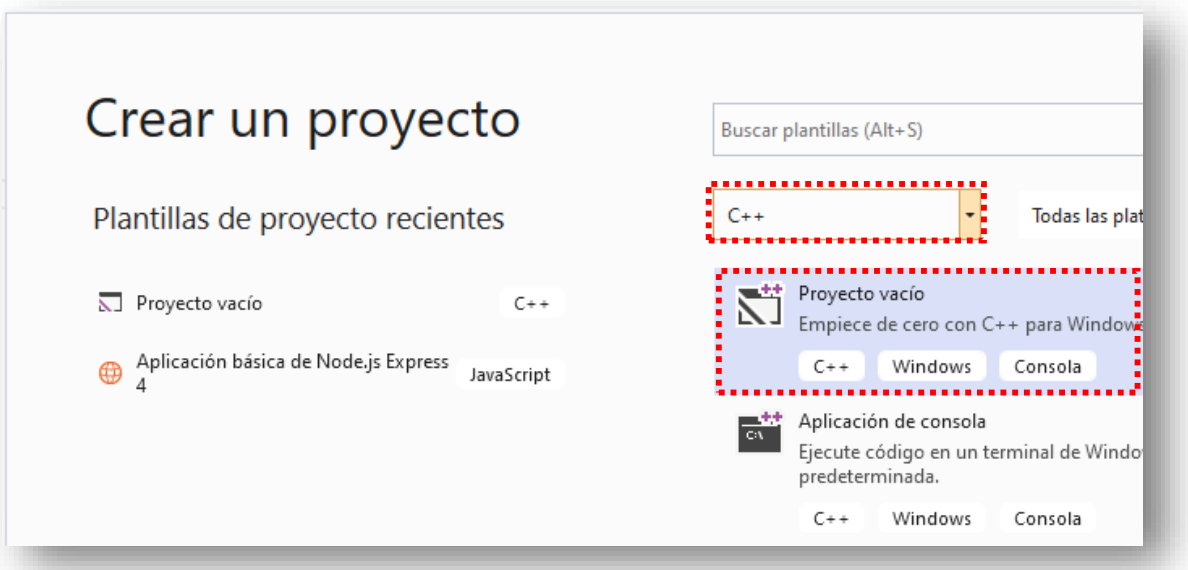


*Figura 2. Cuadro de selección de tecnología y tipo de proyecto.*

4. Descargar el código disponible en la plataforma de docencia virtual en la sección "**Práctica 1**" en carpeta de "**Material de la práctica 1**".
5. Incluir en el primer proyecto el fichero `cliente.c`, y en el segundo proyecto el fichero `servidor.c`. En ambos proyectos se debe incluir también el fichero de cabecera `protocol.h`. Se debe tener en cuenta que ambos proyectos necesitarán la biblioteca **Ws2_32.lib**.
    a. Para incluir un archivo en un proyecto se recomienda copiarlo en la carpeta donde ha creado el proyecto para tenerlo localizable (Visual Studio no obliga a esto, pero así será más fácil de encontrar para copia).
    b. Para añadir un archivo al proyecto se puede pinchar en el explorador de archivos, arrastrar y soltar sobre el nombre del proyecto en el "Explorador de soluciones" (ver Figura 3). También se puede hacer clic

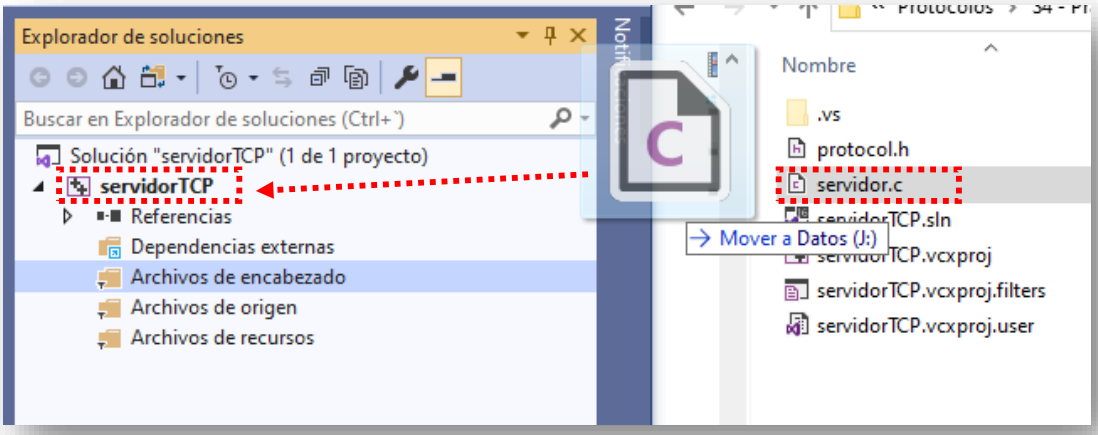


*Figura 3. Añadir ficheros a un proyecto desde el explorador de archivos.*

con el botón derecho en el nombre del proyecto y en el menú que

aparece pinchar en la opción “Agregar” y de nuevo pinchar en “Elemento existente…”.

6. Compilar y ejecutar ambas instancias y comprobar cómo evolucionan el cliente y el servidor.
   a. Si solamente se quiere compilar o enlazar revisar las opciones de la barra de menú en “Compilar”.
   b. Para ejecutar en depuración (como siempre lo haremos en las prácticas) es la opción “Depurar” de la barra de menú. También en la barra de herramientas a través del botón “reproducir” (ver Figura 4).

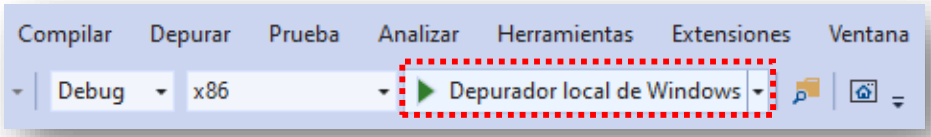


*Figura 4. Botón para ejecutar el proyecto en la barra de tareas.*

Objetivos

1. Conseguir crear dos proyectos correctamente en Microsoft Visual Studio.
2. Conseguir añadir los ficheros de código y compilar ambos proyectos.
3. Conseguir que ambas aplicaciones se ejecuten e intercambien información.

## Tarea 2. Estudio el funcionamiento y la estructura del código

Se deberá analizar el código, tanto del cliente (`cliente.c`) como del servidor (`servidor.c`) para comprender su funcionamiento, tanto a nivel de transporte como de aplicación y además realizar las siguientes tareas:

1. Crear dentro de ambos ficheros comentarios en línea que comiencen con **`//SOCKET`** para cada una de las primitivas Sockets utilizadas y comentar su función dentro de cada aplicación. Añada también la función más relevante de cada argumento de cada primitiva (no repita esto en el caso de que una primitiva aparezca más de una vez en uno de los archivos o en ambos).
2. Crear **solamente dentro del código del servidor** comentarios en línea que comiencen con **`//APLICACIÓN`** en los que se recoja qué función tienen cada uno de los comandos de aplicación que se intercambian el cliente y el servidor. Incluya también el formato que debe tener cada comando con sintaxis ABNF.

Objetivos

1. Conseguir identificar todas las primitivas Socket empleadas en el cliente y el servidor.
2. Conseguir identificar y comentar qué función tienen todos los comandos intercambiados por el cliente y el servidor.

## Tarea 3. Aplicaciones Sockets con TCP

Añada todo el código necesario, tanto al cliente como al servidor, para soportar adecuadamente **los errores de todas las primitivas de transporte** y así conseguir que ambos programas evolucionen de manera predecible. Puede decidir salir del

programa, previa información al usuario y liberación de recursos, en el caso de que se produzca un error.

Objetivos

1. Identificar todas las primitivas de transporte que no tienen un soporte adecuado a los errores que se puedan producir en la comunicación.
2. Incluir en el código las instrucciones necesarias para conseguir que, tanto el cliente como el servidor, evolucionen de forma predecible ante un error de transporte.

## Tarea 4. Modificar el cliente TCP

Modificar el cliente TCP para que permita solicitar al servidor la resolución de la ecuación de segundo grado para coeficientes enteros. Esto se llevará a cabo a través de un mensaje denominado **<EQ2D>**. El cliente deberá solicitar al usuario, una vez autenticado, los números enteros correspondientes a los coeficientes de la ecuación, siendo **a** el coeficiente del término cuadrático, **b**, el coeficiente lineal y **c** el término independiente según la ecuación $\mathbf{ax^2 + bx + c = 0}$. Todos los coeficientes deberán estar entre -99 y 99 por lo que se incluirán las comprobaciones necesarias para que el cliente no envíe valores fuera del rango permitido. El cliente deberá permitir enviar cuantos mensajes de este tipo desee el usuario, por lo que se deberá introducir el código necesario para que la interfaz de usuario evolucione adecuadamente. El formato del mensaje **<EQ2D>** se ha especificado en ABNF y deberá ser exactamente el siguiente:

```
EQ2D = "EQ2D" SP COEF SP COEF SP COEF CRLF; A, B y C
COEF = ["-"]2DIGIT
```

***Ayuda:***

- *Emplear la función `sscanf_s` para leer datos desde teclado, la cual le permitirá leer con un formato y tipo de dato concreto.*
- *Usar la función `sprintf_s` para formar el mensaje a enviar al servidor.*

Objetivos

1. Modificar la aplicación cliente para incluir la captura desde teclado de números (o cadenas con números).
2. Comprobar que los datos ingresados por teclado cumplen con lo especificado por el protocolo.
3. Incluir el código necesario para formar un mensaje de aplicación según la especificación de la tarea.
4. Incluir el código necesario para enviar y recibir el mensaje sobre TCP.
5. Incluir el código necesario para que el usuario pueda enviar cuantos mensajes quiera.

## Tarea 5. Modificar el servidor TCP

Modificar el servidor para que en el caso de recibir el mensaje **<EQ2D>** según el formato de la Tarea 4, el servidor responda al cliente con un mensaje **<OK>** seguido de las dos soluciones de la ecuación con cuatro decimales como mucho. Si el comando no existe, está mal formado, o no se cumplen las condiciones en cuanto a la limitación de longitud de los parámetros, se enviará un mensaje **<ERROR>** con un código de error adecuado. El mensaje **<OK>** incorpora el soporte a las soluciones imaginarias. El cliente siempre deberá mostrar lo que le devuelve el servidor. El formato de las respuestas **<OK>** y **<ERROR>** se ha especificado en ABNF y deberá ser exactamente el siguiente:

```
OK = "OK" SP RESULT SP RESULT CRLF
RESULT = REAL / IMAGINARY CRLF; usar <REAL> si la
                ; solución es un número real e <IMAGINARY>
                ; si la solución es un número imaginario.
REAL = ["-"] 2DIGIT "." 4DIGIT CRLF
IMAGINARY = "IMAGINARIO"

ERROR = "ER" SP CODE CRLF
CODE = DIGIT; 1 Comando erróneo en este estado o no existe
             ; 2 Error en el formato del comando EQ2G
             ; 3 Parámetros fuera de rango.
```

***Ayuda:***

- *Usar la función `sscanf_s` para la lectura de los parámetros recibidos en el mensaje del cliente.*
- *Usar la función `sprintf_s` y las especificaciones de formato para permitir generar los valores de respuesta con las condiciones del protocolo.*

### Objetivos

1. Modificar la aplicación servidora para incluir el análisis del nuevo mensaje de aplicación.
2. Comprobar que los datos recibidos desde el cliente cumplen con lo especificado por el protocolo.
3. Incluir el código necesario para analizar el mensaje de aplicación según la especificación de la tarea y generar una respuesta.
4. Incluir el código necesario para enviar y recibir el mensaje sobre TCP.
5. Incluir el código necesario para el usuario pueda enviar cuantos mensajes quiera.

## Tarea 6. Análisis del protocolo creado

Una vez completada la tarea cinco, ejecute el cliente y el servidor en máquinas diferentes dentro de la misma red de área local (LAN) y capture con la aplicación **Wireshark** el tráfico generado por un intercambio completo entre el cliente y el

servidor que incluya una solución con resultados reales y otra con resultados imaginarios. Filtre los resultados para que tan **solamente** aparezcan los segmentos TCP implicados en la comunicación entre el cliente y servidor.

Esta captura se deberá guardar un archivo de nombre **captura.pcapng** para luego entregarlo como parte de la práctica.

El valor numérico de ejemplo a emplear en los coeficientes será a partir del DNI del/la estudiante de la siguiente manera, teniendo en cuenta que son 8 números: xxAABBCC. Si la práctica se realiza en pareja, será el DNI del/la estudiante con un número de DNI inferior), así para un DNI 25678765 la operación a enviar será: `EQ2D 67 87 65CRLF`.

Objetivos

1. Analizar tráfico con aplicaciones especializadas.
2. Ser capaz de filtrar el tráfico en base a criterios específicos.

## Objetivos de aprendizaje

## Competencias a desarrollar

- **CB2.** Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.
    - *Se aplicarán los conceptos estudiados en teoría al diseño e implementación de clientes y servidores TCP.*
    - *Se explicarán y justificarán las decisiones tomadas por parte del alumnado en cuanto al diseño e implementación de clientes y servidores TCP.*
- **CB3.** Que los estudiantes tengan la capacidad de reunir e interpretar datos relevantes (normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética.
    - *Con el desarrollo de la práctica se persigue la identificación de las características más comunes del funcionamiento de clientes y servidores TCP y de los problemas que se pueden presentar y las maneras de afrontarlos y solucionarlos.*
- **CB4.** Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.
    - *Se dará a conocer terminología específica que permite explicar con precisión y sencillez los conceptos relacionados con las aplicaciones de red sobre TCP.*
    - *Se incluye como parte de la evaluación una entrevista personal para validar los conocimientos adquiridos durante la realización de la práctica.*
- **CB5.** Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.

    - *Se busca que el alumnado pueda aplicar de forma autónoma los conocimientos adquiridos para completar el trabajo complementario necesario para completar las tareas de la práctica.*
- **CG4.** Capacidad de resolver problemas con iniciativa, toma de decisiones, creatividad, y de comunicar y transmitir conocimientos, habilidades y destrezas, comprendiendo la responsabilidad ética y profesional de la actividad del Ingeniero Técnico de Telecomunicación
    - *Se persigue que, planteando un problema a través de una especificación formal, el alumnado pueda llegar a una solución teniendo en cuenta, al menos, consideraciones de eficiencia en el ahorro de recursos.*
- **CG.9.** Capacidad de trabajar en un grupo multidisciplinar y en un entorno multilingüe y de comunicar, tanto por escrito como de forma oral, conocimientos, procedimientos, resultados e ideas relacionadas con las telecomunicaciones y la electrónica.
    - *La explicación incluirá expresiones típicas en inglés, así como recursos de consulta en dicho idioma.*
    - *Se incluye como parte de la evaluación una entrevista personal para validar los conocimientos adquiridos durante la realización de la práctica.*
- **TEL1.** Capacidad de construir, explotar y gestionar las redes, servicios, procesos y aplicaciones de telecomunicaciones, entendidas éstas como sistemas de captación, transporte, representación, procesado, almacenamiento, gestión y presentación de información multimedia, desde el punto de vista de los servicios telemáticos.
    - *Con la práctica se presentan los fundamentos para la creación de un servicio básico sobre el protocolo TCP para Internet.*
- **TEL4.** Capacidad de describir, programar, validar y optimizar protocolos e interfaces de comunicación en los diferentes niveles de una arquitectura de redes.
    - *Con la práctica se presentan los fundamentos para la creación de un servicio básico sobre el protocolo TCP para Internet a través de la definición de un protocolo simple de aplicación.*
    - *Se aplicarán los conceptos estudiados en teoría al diseño e implementación de clientes y servidores TCP.*
    - *Se detallarán las decisiones tomadas por parte del alumnado en cuanto al diseño e implementación de clientes y servidores TCP.*
- **TEL7.** Capacidad de programación de servicios y aplicaciones telemáticas, en red y distribuidas.
    - *Se aplicarán los conceptos estudiados en teoría al diseño e implementación de clientes y servidores TCP.*
    - *Se detallarán las decisiones tomadas por parte del alumnado en cuanto al diseño e implementación de clientes y servidores TCP.*

## Resultados de aprendizaje

La realización de la presente práctica persigue la consecución de los siguientes resultados:

- **Resul-2.** El alumno podrá abordar la **resolución del problema de intercomunicación entre procesos** que se ejecutan en máquinas conectadas utilizando una red de comunicaciones.
- **Resul-3.** Se aprenderá a diferenciar, considerando las características de un protocolo de transporte, **cuál resulta más conveniente utilizar según los servicios de telecomunicación habituales**.
- **Resul-4.** El alumno dominará los protocolos más utilizados en la actualidad para resolver la interconexión de redes de diferente naturaleza, **estableciendo la diferenciación entre los protocolos vinculados al transporte de información**, los relativos al intercambio de información para el encaminamiento y los auxiliares a los dos tipos descritos anteriormente.
- **Resul-26.** Adquirir **facilidad para el manejo de especificaciones, reglamentos y normas** de obligado cumplimiento.
- **Resul-27.** Resolver **problemas con iniciativa, toma de decisiones y creatividad**.
- **Resul-30.** Adquirir **facilidad para el manejo de especificaciones, reglamentos y normas** de obligado cumplimiento.
- **Resul-34.** Trabajar en **un grupo multidisciplinar y en un entorno multilingüe**.
- **Resul-35.** Comunicar, tanto por escrito **como de forma oral, conocimientos, procedimientos, resultados e ideas relacionadas con las telecomunicaciones** y la electrónica.

# Detalles del ejercicio

## Objetivos generales

La realización de la presente práctica persigue la consecución de los siguientes objetivos:

- Familiarización con las primitivas socket.
- Creación de aplicaciones cliente sencillas basadas en sockets.
- Creación de aplicaciones servidor sencillas basadas en sockets.
- Familiarización con el entorno de desarrollo de aplicaciones usado en el curso.
- Familiarización con los recursos de referencia de Internet.
- Familiarización con la sintaxis ABNF.
- Familiarización con el software de análisis Wireshark.

## Recursos

- Ordenador PC.
- Conexión a Internet.
- RFCs de la IETF (https://www.rfc-editor.org/)
- Documentos y material de estudio de la asignatura.
- Microsoft Visual Studio Community Edition.

## Programación

La presente práctica se programa en tres sesiones de 2 horas en el laboratorio (6 horas de trabajo en clase/9 horas de trabajo autónomo), distribuidas de la siguiente manera:

- Primera sesión (2 horas):
  - Tarea 1. Introducción a la plataforma y programas Sockets.
  - Tarea 2. Estudio el funcionamiento y la estructura del código.
  - Tarea 3. Aplicaciones Sockets con TCP.
- Segunda sesión (2 horas):
  - Tarea 4. Modificar el cliente TCP.
- Tercera sesión (2 horas):
  - Tarea 5. Modificar el servidor TCP.
  - Tarea 6. Análisis del protocolo creado.

## Material a entregar

Se deberá subir en la tarea reservada para la entrega en la plataforma de docencia de la UJA un **fichero comprimido** con el siguiente contenido:

- Código fuente: añadir todos aquellos ficheros de **código fuente en C** necesarios para la adecuada compilación y construcción del código (.c y/o .h), **nunca los ficheros del proyecto, ficheros ejecutables o de código objeto**.
  - El código entregado deberá cumplir con los objetivos concretos descritos en la práctica, tener comentarios que ayuden a su seguimiento y corrección, además de no tener errores de sintaxis, ni de ejecución.
- Fichero **captura.pcapng** con la captura de Wireshark adecuadamente filtrada según lo indicado en la Tarea 6.

Además, el fichero comprimido a entregar deberá cumplir con lo siguiente:

- El nombre del fichero deberá comenzar por **práctica1_** y estar seguido por el identificador de quien/quiénes hayan realizado la práctica. Ejemplo: **práctica1_estudiante1** o **práctica1_estudiante1_estudiante2** para dos estudiantes.
- En el caso de que la práctica se realice en pareja, **cada estudiante** deberá entregar el mismo material por separado en la plataforma de docencia de la UJA.

El no cumplir con los requisitos de entrega podrá conllevar una penalización en la calificación final de la práctica de hasta 1 punto.

## Evaluación

La evaluación de esta práctica, una vez obtenido el visto bueno del profesor/a tras una entrevista individual o en grupo, según el caso, se hará en función de la rúbrica publicada para esta práctica.

Si se entrega una práctica que no se pueda ejecutar, ya sea por errores de compilación o enlazado, esto podrá conllevar, desde una penalización en la nota final de la práctica de como mínimo un punto, a la calificación de 0 puntos en el total de la práctica.

Téngase en cuenta que la copia, o la no obtención del visto bueno del profesor/a, conllevará la calificación de 0 en la práctica.

## Bibliografía recomendada

1. Donahoo, M. J. "**TCP/IP sockets in C**" 2ª Edición, Morgan Kaufmann.
2. Crocker, D., Ed., and P. Overell, **"Augmented BNF for Syntax Specifications: ABNF"**, STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <http://www.rfc-editor.org/info/rfc5234>.