



# Redes de Computadores – Laboratorio 1: Transferencia de archivos por TCP

28 de abril de 2022

Para este laboratorio, tendrán que crear un programa que permita la transferencia de archivos entre dos computadores, el cual descompondremos en dos mitades. Para esta primera mitad, implementarán este programa usando *sockets* TCP, pudiendo usar el lenguaje de su preferencia, mientras trabajen directamente con bibliotecas de *sockets* y no utilicen soluciones completas como `libcurl`, que ya implementan protocolos de transferencia de archivos.

## 1. Requerimientos

Los requerimientos a cumplir son los siguientes:

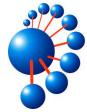
- El programa debe utilizar conexiones TCP de forma manual, trabajando directamente con *sockets*.
- El programa debe poder ejecutarse en modo receptor o transmisor (pueden ser ejecutables diferentes o uno solo, indicando cuál modo usar mediante argumentos).
- El protocolo debe permitir la transferencia de archivos (binarios o de texto plano) de cualquier tamaño, así como también indicar el nombre y tamaño del archivo a transferir, para que pueda ser llamado de la misma forma en su destino.
- Ambas partes deben indicar el progreso de la transferencia a medida que el archivo se envía.
- El protocolo debe permitir la transferencia cifrada de archivos mediante algún protocolo de cifrado simétrico (cifrado por contraseña).

## 2. Evaluación

El laboratorio se evaluará de a pares bajo los siguientes requerimientos, mediante una escala de 6 puntos:

- **Funcionalidad** (2 puntos): El programa debe permitir transferir archivos de forma exitosa y confiable, señalizando errores en caso de haberlos.
- **Transferencias de tamaño arbitrario** (1 punto): El programa debe poder transmitir archivos de cualquier tamaño.
- **Información sobre la transferencia** (1 punto): El programa debe poder indicar el progreso de la transferencia, además del nombre y tamaño del archivo a su receptor.
- **Cifrado** (1 punto): La transferencia debe poder realizarse mediante algún tipo de cifrado simétrico (como Blowfish o AES).
- **Demostración** (1 punto): El uso del programa debe ser demostrado, mostrando que es posible la transferencia de archivos, tanto de forma cifrada como sin cifrar.

De forma adicional, cada uno de estos requisitos extras otorgarán 0.5 puntos adicionales sobre la calificación anterior, hasta completar 6 puntos:



- **Limpieza de código:** El código del programa tiene una sintaxis clara, consistente y limpia.
  - **Pista:** Pueden utilizar programas de formato automático como `clang-format` (C/C++), `autopep8` (Python) o `rustfmt` (Rust), dependiendo del lenguaje que quieran utilizar. Este clase de utilidades pueden ser integradas fácilmente en editores de código como Visual Studio Code o (Neo)vim.
- **Compilación automatizada:** El programa utiliza un *toolchain* de compilación, que se encarga de ejecutar los pasos de compilación de forma automática
  - **Pista:** Dependiendo del lenguaje, pueden utilizar herramientas como CMake, `autotools` (sólo para valientes) o Cargo.
- **Cifrado asimétrico:** El programa permite utilizar algún algoritmo de cifrado asimétrico para transferir archivos, como RSA.

El código debe ser entregado como un archivo comprimido (de cualquier tipo) o un *link* a un repositorio de código, junto con un **README** con instrucciones de cómo compilarlo y ejecutarlo. También se requerirá un video corto de demostración (no mayor a 5 minutos) mostrando el funcionamiento del programa, para el cual deben entregar un *link* (consejo: para hacer más fácil la grabación, pueden ejecutar cliente y servidor en `localhost`; es igual de válido).