

# Protocolos De Internet

## Trabajo Práctico 1 - Parte C - 2)

A continuación explico el código del programa escrito. Basicamente, lo único que tuvimos que cambiar fue el servidor ya que el cliente es el mismo. Ejecutamos `./TCP_server` y el servidor se encontraba disponible para poder aceptar conexiones. Pero, a diferencia de los primeros ejercicios, acá usaremos la función `select()`, lo cual en el código adjuntado a este archivo se encuentra implementado de la siguiente manera:

```
select(maxFileDescriptor+1,&readFileDescriptorsSet,NULL,NULL,NULL);
```

Esto nos permite a nosotros poder usar el `accept()`, o `recv()` sin bloqueo ya que nos dice en que sockets ya podemos realizar una de estas funciones sin bloqueo. En otras palabras, los clientes mandaron respuesta. Como primer parámetro le pasamos el descriptor de archivo más grande que tenemos y luego la dirección del arreglo (aca hablo del tipo de dato `fd_set` en donde ponemos los sockets a las cuales nosotros esperamos respuesta. Luego `select()` lo modifica de manera que nos deja aquellos sockets en la cual efectivamente tenemos algo para leer. Luego, una vez realizada la lectura sobre ese socket, reenviamos lo obtenido a todos los otros clientes conectados. ¿Donde se encuentran estos clientes? En el array del tipo `fd_set` que fuimos modificando a medida que iban llegando clientes (es un array totalmente distinto al paso por parámetro en `select`). En el código se encuentra implementado de la siguiente manera:

```
fd_set fileDescriptorsSet;
```

Para poder registrar el tiempo de los clientes se utilizó `time()`