

## Correcciones

1. Descomenté algunos mensajes de error mientras que otros los eliminé por completo.
2. Hice correcciones del código, como por ejemplo arreglar los paréntesis en las macros.
3. Se modificó la función `socket_tcp_enviar` dado que a pesar de tener un `while` para asegurar de enviar todo el mensaje no se movía en el buffer con el fin de enviar en el siguiente intento los bytes restantes. Con la modificación en caso de no enviar la totalidad del mensaje (es decir, los enviados eran menor a la longitud) en el siguiente intento se envía un puntero desde la cantidad de enviados:

```
send(self->fd, &buff_nuevo[enviados], longitud-enviados, MSG_NOSIGNAL);
```

4. Respecto al protocolo, eliminé que el servidor tenga que mandar un byte mas para dar cuenta del `'\0'` del fin de cadena y modifiqué para que lo agregue el cliente dado que este el que sabe que está recibiendo y para qué lo va a usar. A fines prácticos, el cliente malloquea para el tamaño del mensaje a recibir +1, y luego al final agrega un `'\0'`.

## Modificaciones posibles pero no realizadas (por falta de tiempo)

**a.** Para dar cuenta de la corrección sobre que el TDA *sector* es muy complejo. No tuve tiempo de modificarlo, pero se me ocurren dos modificaciones que se le podría hacer:

1. En primer lugar, que se guarden las celdas en el stack en vez de memoria dinámica.
2. En segundo lugar, para hacerlo un poco mas flexible, el sector podría recibir en su constructor, la capacidad. De manera que se vuelve un TDA ligeramente mas reutilizable.

**b.** Hacer un array de array para las celdas del tablero, aunque es mas complejo que hacer un array unidimensional a mi me permitió acceder por fila y columna mas fácilmente. Lo que podría ser un cambio posible, dado que el tablero de un sudoku no es muy grande declararlo en el stack de la siguiente manera:

```
celda_t celdas[9][9];
```

Otra opción declararlo parcialmente en el stack guardando el vector de filas allí y cada fila en el heap.

**c.** Finalmente sobre el comentario de si es necesario un TDA *celda*. Una alternativa posible es eliminar el TDA *celda* y que las pocas validaciones que esta hace se hagan en el TDA *tablero*, que es el que maneja toda la lógica del sudoku. Esto se podría lograr, haciendo que el TDA *tablero* tenga 2 “tableros”, uno con los valores numéricos de cada fila y otro con un valor booleano que indica si esta es inmutable o no.