

Descripción de la solución

Para la solución del problema propuesto se procedió a diseñar una solución basada en TDA's y una librería para la función que representa el trabajo del código genético. Se modelaron los distintos entes del problema en forma de TDA's, cada uno con una responsabilidad definida y en colaboración con los demás, pero sin depender de su implementación.

Para la implementación de las distintas constantes del problema como las tablas de transformación codón - aminoácido y la codificación de las bases nitrogenadas se harcodeo en el código estas constantes tratando de encapsularlas lo más posible dentro de sus respectivas abstracciones.

Para la comunicación cliente - servidor se utilizó un protocolo basado en TCP que consiste en el envío secuencial de los bytes correspondientes a cada codón leído en el archivo origen del cliente y el cierre de la conexión en ese sentido por parte del cliente. Detectado este cierre el servidor procede a enviar el mensaje generado y cierra la conexión finalizando la comunicación entre el cliente y el servidor.

Para el manejo de los sockets se creó un TDA que encapsula la utilización de funciones del sistema operativo y el manejo de errores de estas. Este TDA provee una interfaz simple y genérica para el uso de sockets tanto en modo servidor como en modo cliente y para el envío y recepción de datos.

Tanto el envío como la recepción de bytes se realiza mediante un buffer de un tamaño de 20 bytes tanto para el cliente como para el servidor. La lectura del archivo de entrada por parte del cliente se realiza de a 60 bytes. Con esto se reducen los llamados a las funciones de lectura de un archivo y de envío y recepción de datos a través del socket, que son la parte crítica en la performance de la aplicación. La elección de los tamaños de buffer fue arbitraria salvo por que la lectura desde el archivo se realiza en un numero múltiplo de tres para facilitar la obtención de los codones correspondientes. Luego de recibidos los codones codificados y transformados en aminoácidos a través del código genético, el servidor cuenta para cada tipo de aminoácido la cantidad recibida. Esto se implementó con un array del tamaño de la cantidad de aminoácidos máximo (esta constante esta declarada en el TDA aminoácido) que a cada posición se le asignó un aminoácido distinto a través del método del TDA aminoácido que retorna un número para cada aminoácido. De esta forma se logró facilitar la implementación de la cuenta y esclarecer el código.

Para el ordenamiento de los datos en el servidor se utilizó la función predefinida `qsort` en `stdlib`. La comparación se realiza entre las estructuras auxiliares de conteo, si la cantidad contada es igual se procede a comparar los nombres de los aminoácidos correspondientes. Para esta comparación se utiliza una función de comparación de contadores y una de aminoácidos definida en su correspondiente TDA. Esta última guarda para la comparación la representación de los aminoácidos en formato cadena sin tildes. Pero para la impresión se utiliza el método `to_string` que retorna el string correspondiente con los tildes.

Esquema de diseño

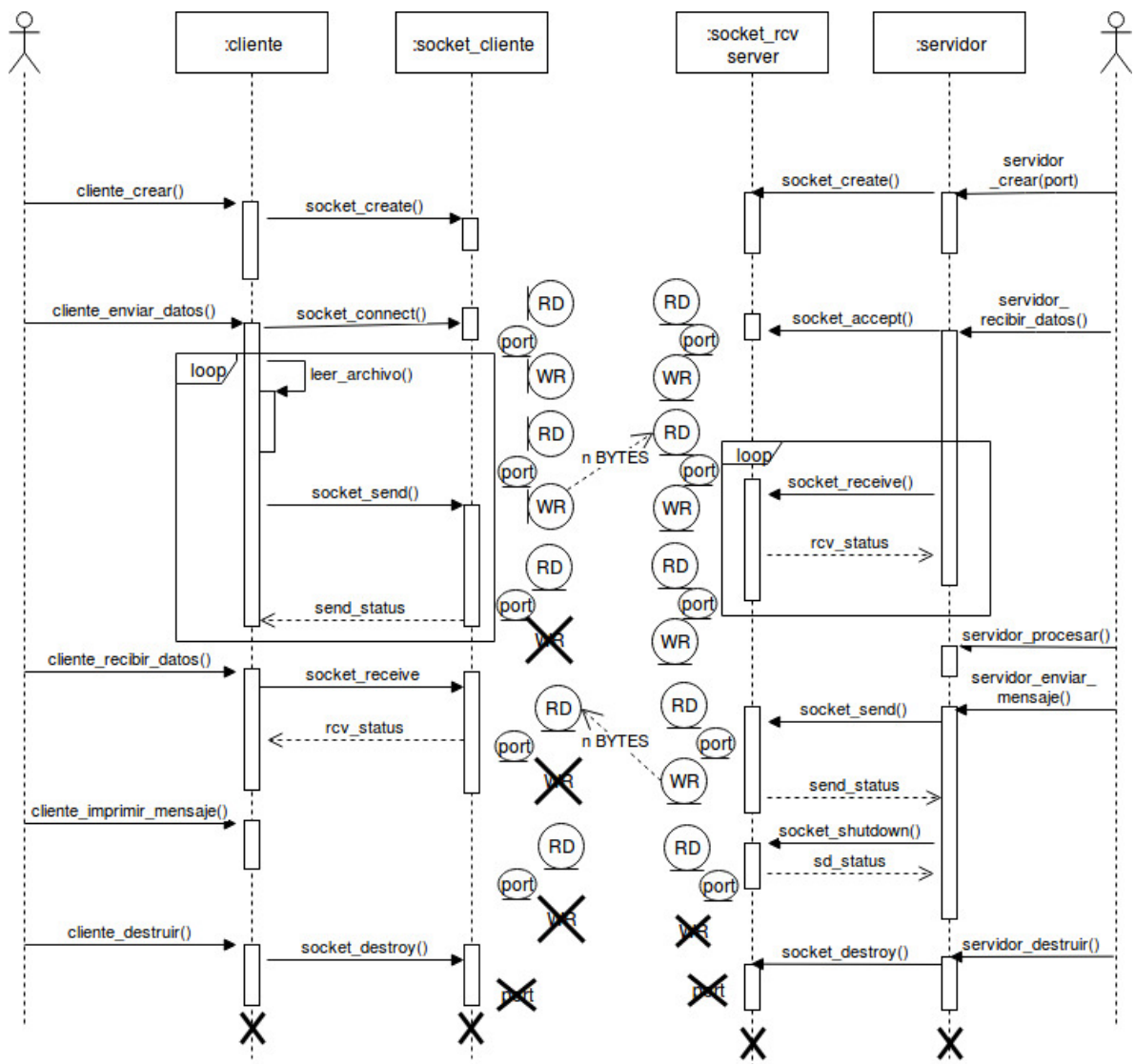


Figura 1: Diagrama de secuencia del envío y recepción de datos.