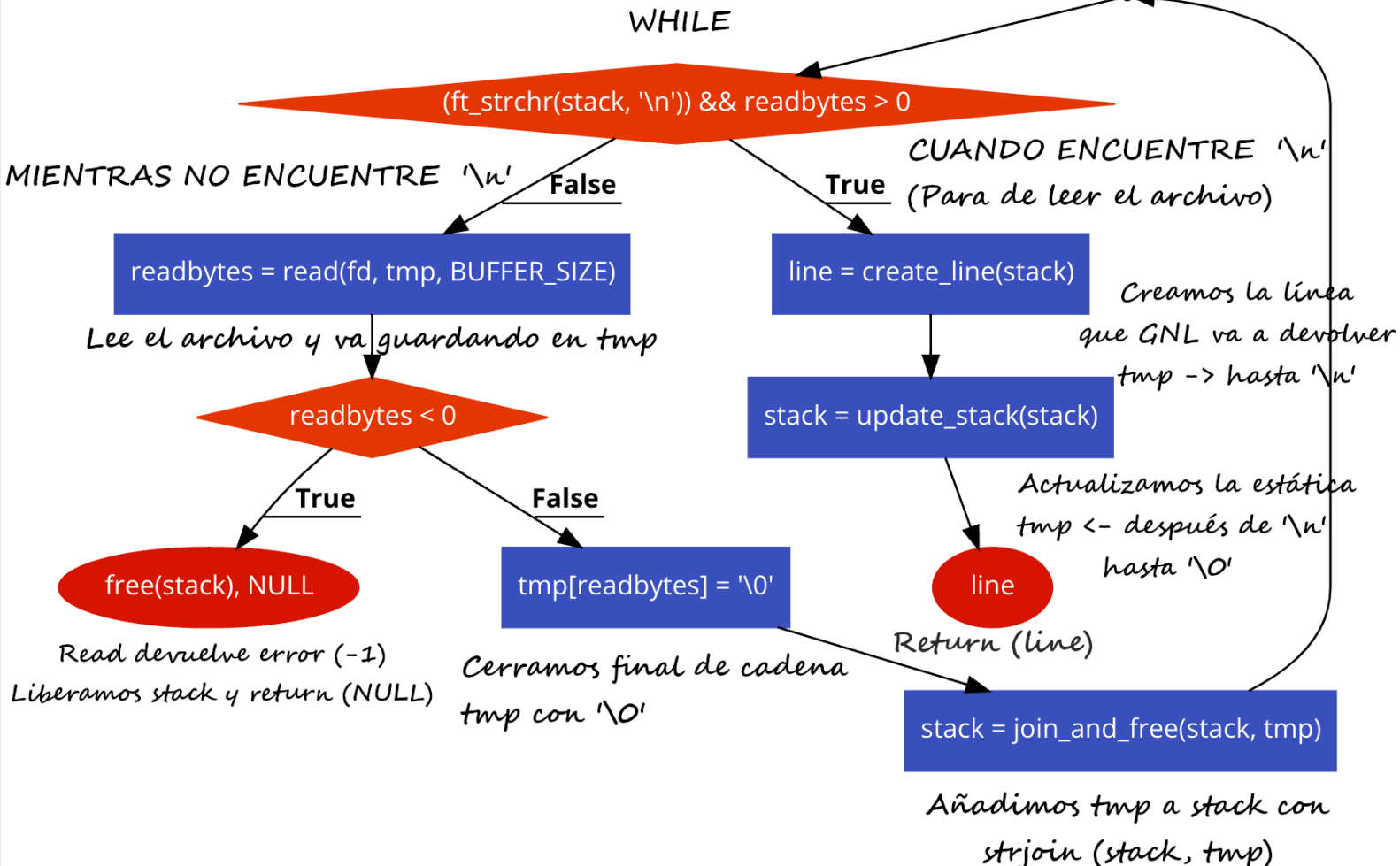


Desde main enviamos el file descriptor (fd)

En get_next_line declaramos las variables:

```
static char *stack = NULL;  
char tmp[BUFFER_SIZE + 1];  
char *line;  
ssize_t readbytes;
```



char create_line(char stack)

En la cadena estática tenemos todo lo que se ha leído con read, según el buffer que tuviese puede haber más de un '\n'.
Vamos a copiar de la estática a char *line todo lo que encontremos hasta '\n'

char *line

int i

IF
stack || !*stack

Comprobamos que la cadena estática y su contenido no sean NULL

True

i = 0

Calculamos tamaño de la línea a crear
Mientras no sea '\n' o '\0' -> i++

stack[i] != '\n' && stack[i] != '\0'

True

i++

False Si [i] == '\n' i++

stack[i] == '\n'

True Porque hay que añadir '\n' a la cadena a devolver

i++

line = malloc(sizeof(char) * i + 1)

Asignamos memoria i + 1 (para cerrar con '\0')

line

False

True

i = 0

Reiniciamos el contador, que ahora usamos como índice para ir copiando a char *line

stack[i] != '\n' && stack[i] != '\0'

False

True

line[i] = stack[i]

Vamos copiando
line[i] == stack[i]

i

i++

IF
stack[i] == '\n'

True

line[i++] = '\n'

False

line[i] = '\0' <- Cerramos cadena

Si la última posición de la estática es '\n' también la copiamos a line

line <- Return (line);

line

char update_stack(char stack)

Ya hemos conseguido en create_line la línea que queremos devolver, ahora vamos a hacer que en nuestra estática solo quede el resto de caracteres leídos después de esa línea

Usaremos una cadena auxiliar para así poder liberar la memoria anterior de la estática

(free stack y luego stack = aux)

char *aux

-< Creamos una cadena char auxiliar

char *p

-< Puntero a cadena char*

int i

-< Índice

p = ft_strchr(stack, '\n')

Puntero = posición de la estática donde ft_strchr encuentre '\n'

IF

p

False

True

free(stack), NULL

p++

Desplazamos el puntero una posición más, para "saltarnos" '\n'

Asignamos memoria según len de la estática desde -> la posición del puntero + 1

aux = malloc(sizeof(char) (ft_strlen(p) + 1))

aux

False

True

NULL

i = 0

Copiamos el contenido de la estática en la cadena auxiliar nueva (desde la posición del puntero, que ya sabemos que ha "saltado" la línea creada)

*p != '\0'

True

False

aux[i] = p

aux[i] = '\0'

RETURN (free stack, aux)

free(stack), aux

Importante liberar stack justo antes de terminar y devolver aux

i++

p++