Il funzionamento di quesa coda di messaggi è ispirato al servizio SQS di Amazon ed è stato strutturato nel seguente modo: una volta avviati i 3 eseguibili (queue, producer e consumer) il producer inserirà tramite chiamata RPC alla coda messaggi contenenti un testo di 15 caratteri generato casualmente, se la RPC va a buon fine la queue mette il messaggio in coda con stato INQUEUE, il seguente stato sta a indicare che il messaggio è nella coda ma non è stato ancora inviato, quando il consumer chiamerà la funzione pull del messaggio tramite chiamata RPC la coda invia il messaggio al consumer e setta lo stato del messaggio come SENT, questo sta ad indicare alla coda che il messaggio è stato inviato ma per questo messaggio non è stato ricevuto ancora l'ACK, l'ACK di ricezione sarà inviato dal consumer sempre tramite chiamata RPC alla coda una volta che riceverà il messaggio, nel frattempo la coda ha avviato una go routine che si preoccuperà di controllare ciclicamente se per quel messaggio è stato ricevuto l'ACK, trascorsi TIMEOUT_RETRASMIT (constante modificabile definita in const.go) secondi se l'ack non è stato ancora ricevuto la go routine retterà lo stato di quel messaggio di nuovo come INQUEUE, ovvero da reinviare.
Nel caso l'ACK invece sia arrivato alla coda la go routine terminerà per chiamare a sua volta un'altra go routine che si preoccuperà invece di gestire il TIMEOUT_VISIBILITY (sempre configurabile nel file const.go), ovvero il messaggio potrebbe essere stato ricevuto dal consumer, l'ACK è giunto alla coda ma subito dopo il consumer ha subito un crash quindi il messaggio non è mai stato elaborato dal consumer e quindi bisogna spedirlo di nuovo quindi trascorsi TIMEOUT_VISIBILITY secondi se la coda non ha ancora ricevuto il secondo ACK, ovvero l'elaboration ACK la go routine setterà lo stato del messaggio come INQUEUE, ovvero il messaggio dovrà essere rimesso in coda perché non è mai stato elaborato da un consumer.

La coda grazie ad una attenta gestione dei mutex funziona egregiamente anche sotto pesanti carichi di lavoro, ovvero con diversi producer e consumer in esecuzione nello stesso momento.