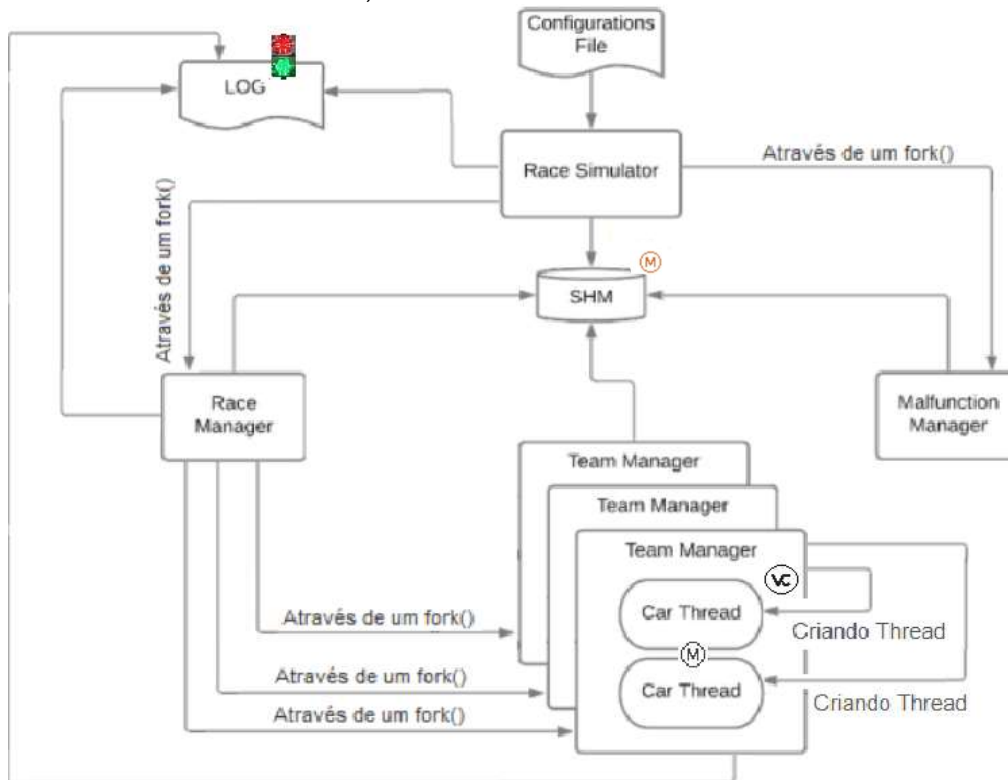


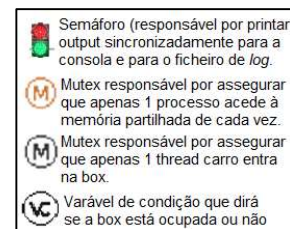
## Projeto SO – Meta 1

Duarte Emanuel Ramos Meneses, 2019216949

Patrícia Beatriz Silva Costa, 2019213995



O programa inicia-se com a abertura do ficheiro de *log* onde serão registadas todas as informações respeitantes à aplicação. Posteriormente, cria-se um semáforo que será utilizado para enviar de forma síncrona o *output* para o ficheiro de *log* e para o ecrã. Caso este semáforo não existisse, corríamos o risco de os prints não serem executados na ordem correta e com a informação pretendida. É criada memória partilhada (*SHM*) para uma estrutura que conterá as informações dos carros e das boxes/equipas. Todos os processos que acedem a esta memória partilhada irão fazê-lo através de um *mutex* (M laranja no diagrama) para que apenas 1 processo possa acedê-la de cada vez. Se isto não acontecesse, por exemplo, se 2 processos acedessem a essa memória ao mesmo tempo, 1 para ir buscar o valor de uma variável e o outro para o atualizar, caso a atualização se efetuasse posteriormente quando devia ser antes, o processo que foi buscar o valor teria acesso a um valor desatualizado.



De seguida, as informações relativas às configurações da corrida são lidas do ficheiro correspondente e guardadas numa estrutura criada para o efeito. Note-se que todas estas operações são realizadas no processo respeitante ao simulador de corrida (*Race Simulator*). A seguir é criado o processo gestor de corrida (*Race Manager*) que irá, entre outras coisas, criar os processos gestor de equipa (*Team Manager*) - 1 por equipa. Em cada gestor de equipa serão criadas *x threads* (*Car Thread*) sendo *x* o número de carros em competição dessa equipa. Na função responsável por gerir as *threads* (carros), existe um *mutex* (M preto no diagrama) uma vez que apenas 1 carro pode estar na box de cada vez (VC, no diagrama, dirá se a box está ocupada ou não – se estiver, os carros saberão e não tentarão entrar para não ficar à espera). Ainda no processo do simulador de corrida é criado outro processo que dirá respeito ao gestor de avarias (*Malfunction Manager*), responsável por gerar avarias nos carros.