Bom dia, infelizmente o final de semana não aconteceu como o planejado, meu superior na Volkswagen solicitou que eu trabalhasse sábado (inclusive me pagou a parte), pois haveria uma atualização no sistema e pediu que eu acompanhasse essa atualização realizando monitoramento constante do sistema de comunicação com fornecedores, dentro da planta Anchieta.

Porém consegui realizar o desafio proposto, exceto a parte de testes onde realmente não tive tempo hábil para realiza-los em Junit e poderia me aprofundar muito mais na orientação a objetos.

O projeto foi desenvolvido com a IDE Eclipse, e minha previsão realização de teste seria com Junit. Não utilizei nenhum framework específico (Java Standard Edition).

Para o perfeito funcionamento do projeto é necessário que o arquivo de log se chame log.txt e esteja localizado no diretório C da máquina (C:), caso exista a necessidade de alterar o path do arquivo, é necessária uma atualização da classe ParseCSV na 16° linha em

BufferedReader BuffReader = **new** BufferedReader(**new** FileReader("c:\\log.txt"));

Onde o novo path pode ser especificado no construtor de FileReader.

Minha aplicação se divide em seis classes, que são RankingPartidaAssassino, RankingPartidaMortos, RankingGeralAssassino, RankingGeralMorto , ParseCSv e ExecutaRanking.

Explicando cada classe.

ParseCSv: Essa classe fica encarregada de extrair o dados do arquivo que se chama log.txt localizado no diretório C:, ela lê linha por linha e monta uma String a qual será processada pelas outras classes.

ExecutaRanking: Ela basicamente instancia um objeto de cada classe e executa todos eles em seu método principal.

RankingPartidaAssassino: Essa classe pega a String que vem de ParseCSV, e realiza o processamento dessa String para montar um placar onde verificamos quem foi o assassino, em qual partida e sua quantidade de assassinatos nessa partida.

RankingPartidaMortos: Essa classe pega a String que vem de ParseCSV, e realiza o processamento dessa String para montar um placar onde verificamos quem foi morto, em qual partida e sua quantidade de mortes.

RankingGeralAssassino: Apesar de não ter sido solicitado no desafio, essa classe pega a String que vem de ParseCSV, e realiza o processamento dessa String para montar um placar geral onde verificamos quem foram os assassinos, sua quantidade de assassinatos em TODAS as partidas realizadas.

RankingGeralMortos: Apesar de não ter sido solicitado no desafio, essa classe pega a String que vem de ParseCSV, e realiza o processamento dessa String para montar um placar geral onde verificamos quem foram os mortos e sua quantidade de mortes em TODAS as partidas realizadas.