

Instruções para entrega: Enviar dois arquivos para *mlc2@cin.ufpe.br* : (1) arquivos .hs e .java, com respostas (copiar e colar os códigos das respostas) e (2) um arquivo no formato pdf com **todas** as respostas. Escrever o nome na primeira linha dos arquivos. O assunto do email deve estar no formato: Nome completo_questaoN_2ee_plc_2018_1. Os nomes dos arquivos devem estar neste mesmo formato

- (2,5) 1. Implemente, em Java, um vetor que seja seguro para uso com threads. Os métodos *get*, *set* e *swap* devem ser implementados.
- (2,5) 2. Implemente uma fila bloqueante em Java. Não usar funções da biblioteca de Java.
- (2,5) 3. O forno de uma padaria tem capacidade para assar 50 pães simultaneamente. À medida que pães ficam prontos, são retirados do forno. O abastecimento, que acontece apenas após o forno ser completamente esvaziado, é feito de maneira que 10 pães são colocados no forno por vez, até a capacidade do forno. Assuma que o primeiro lote de pães colocados no forno é também o primeiro a ser retirado. Utilizando uma fila bloqueante, implemente, em Java, o comportamento dessa padaria. Considere que retirar os pães do forno é um processo mais lento que o abastecimento dele, devido ao tempo necessário para assar os pães.
- (2,5) 4. Utilizando memória transacional, implemente um buffer em Haskell. Defina o tipo *Buffer* que pode guardar valores de um tipo qualquer, uma função para criação de um buffer (tipo: **IO** (Buffer a)) (utilize a função *newTVarIO*), a função *put* (tipo: Buffer a \rightarrow a \rightarrow STM()) e a função *get* (tipo: Buffer a \rightarrow STM a). A função *put* sempre coloca um novo dado no buffer; a função *get* retorna um valor do buffer, porém aguarda se o buffer estiver vazio.