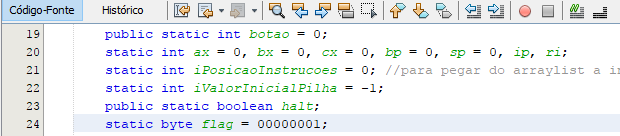
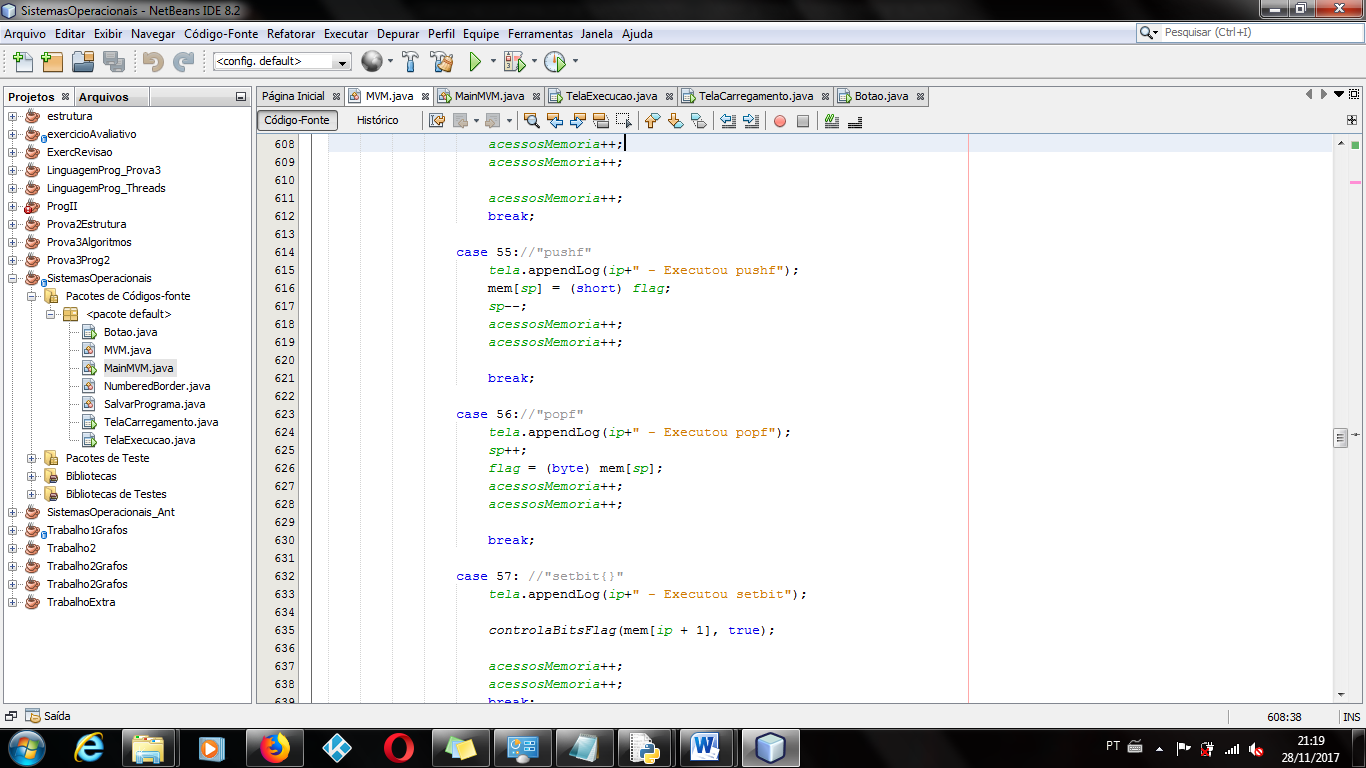
RELATÓRIO MVM

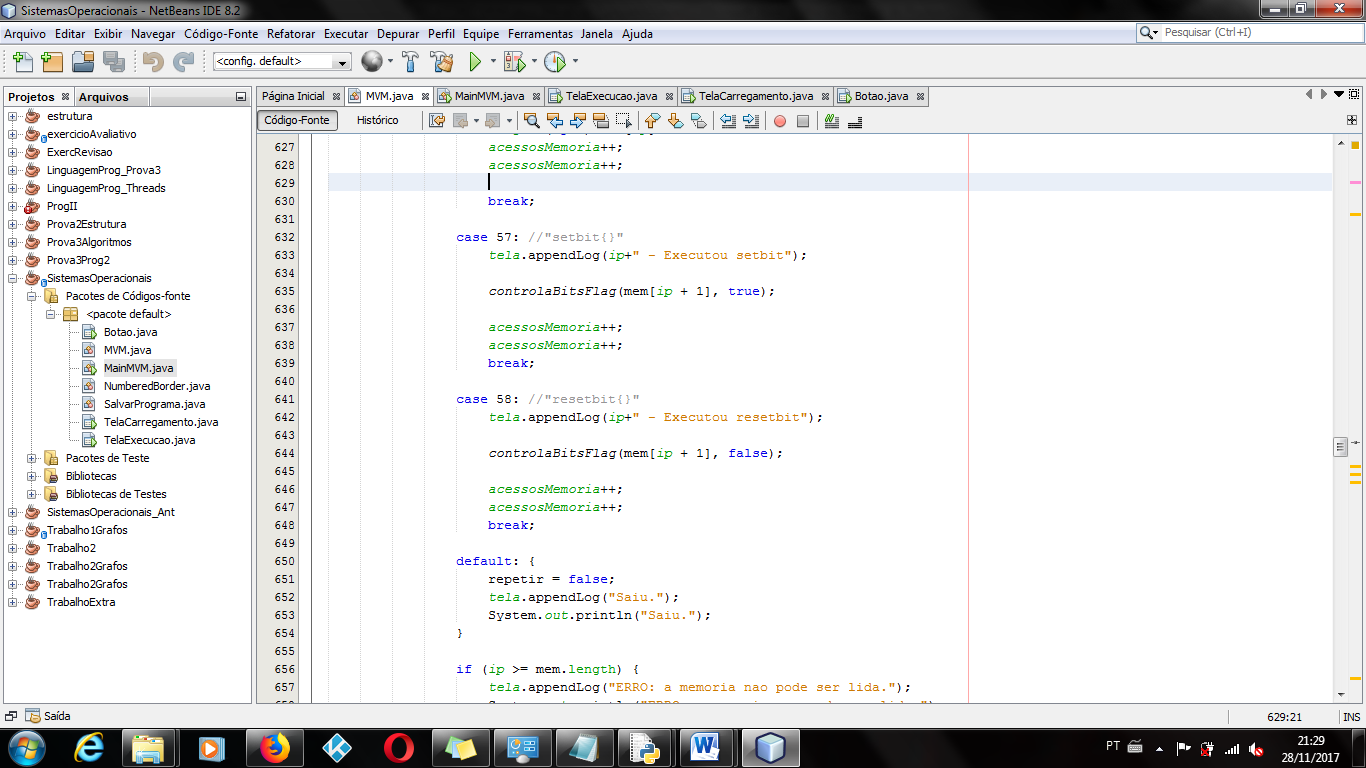
1. Criamos a variável “flag” do tipo byte com inicialização do bit 0 ativo – MVM.java – Linha 24



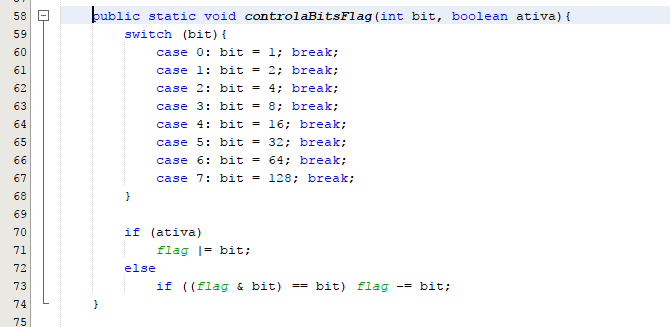
1. Implementamos o funcionamento das funções “pushf” e “popf” que vão empilhar e desempilhar, respectivamente, a flag da memoria – MVM.java – Linha 614



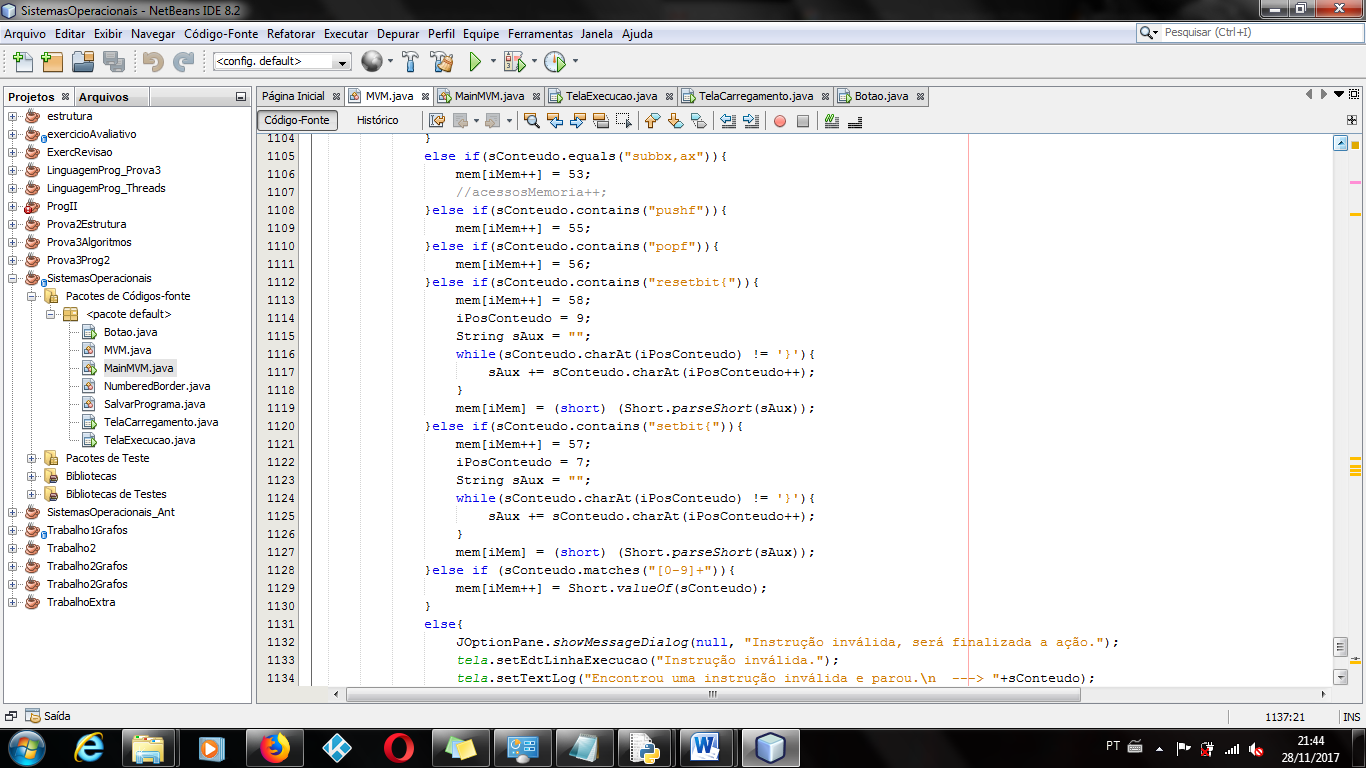
1. Implementamos o funcionamento das funções “setbit{}” e “resetbit{}” que vão ativar e desativar o bit da flag que é passado por parâmetro – MVM.java – Linha 632



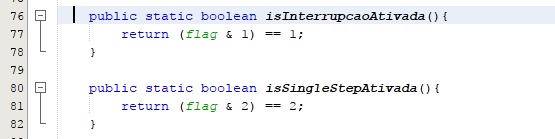
1. Criamos a função “controlaBitsFlag” que vai ativar ou desativar os bits da flag de acordo com os parâmetros. O switch case codifica o bit de acordo com o valor correto e depois ativa ou desativa este bit da flag – MVM.java – Linha 58



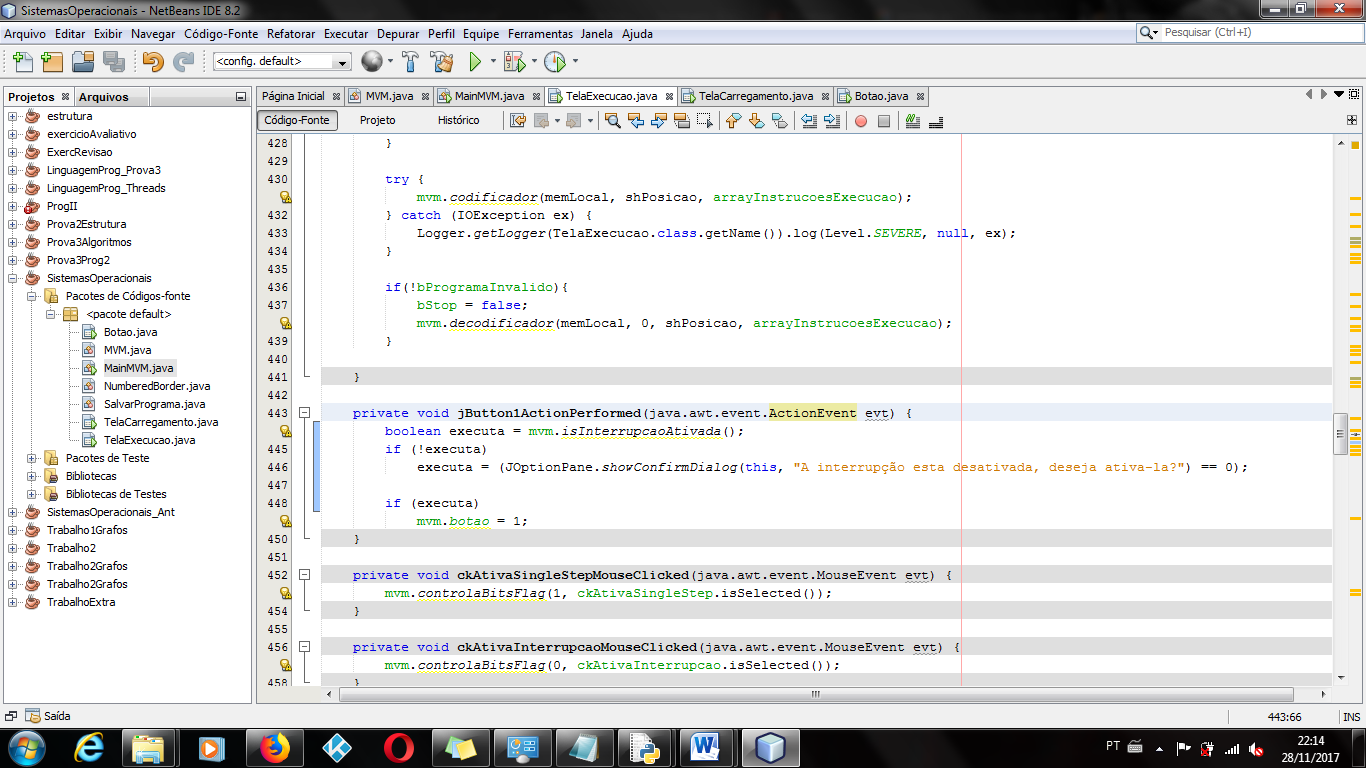
1. Implementamos a codificação das funções “pushf”, “popf”, “setbit{}” e “resetbit{}” – MVM.java – Linha 1105



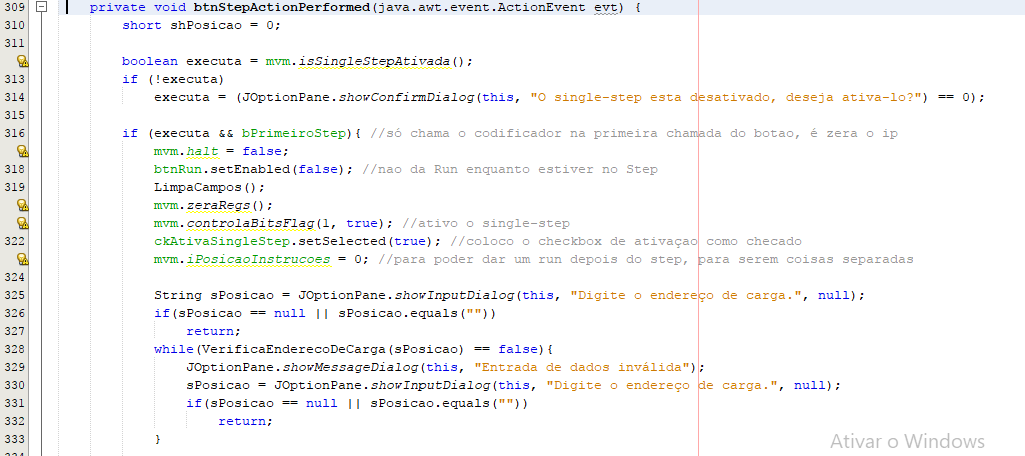
1. Adicionamos as funções “isInterrupcaoAtivada” e “isSingleStepAtivada” para verificar se os bits responsáveis pelo controle da interrupção e do single-step estão ativos – MVM.java – Linha 76



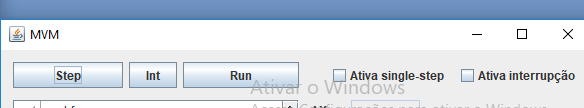
1. Adicionamos a verificação da flag ativa para a interrupção – TelaExecucao.java – Linha 444



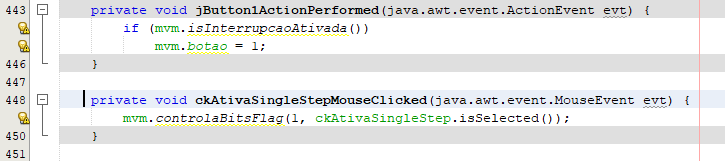
1. Adicionamos a verificação da flag ativa para o single-step – TelaExecucao.java – Linha 312



1. Adicionamos os dois checkbox de ativação da interrupção e do single-step – TelaExecucao.java



1. Implementamos o Controle de ativação do single-step – TelaExecucao.java – Linha 448

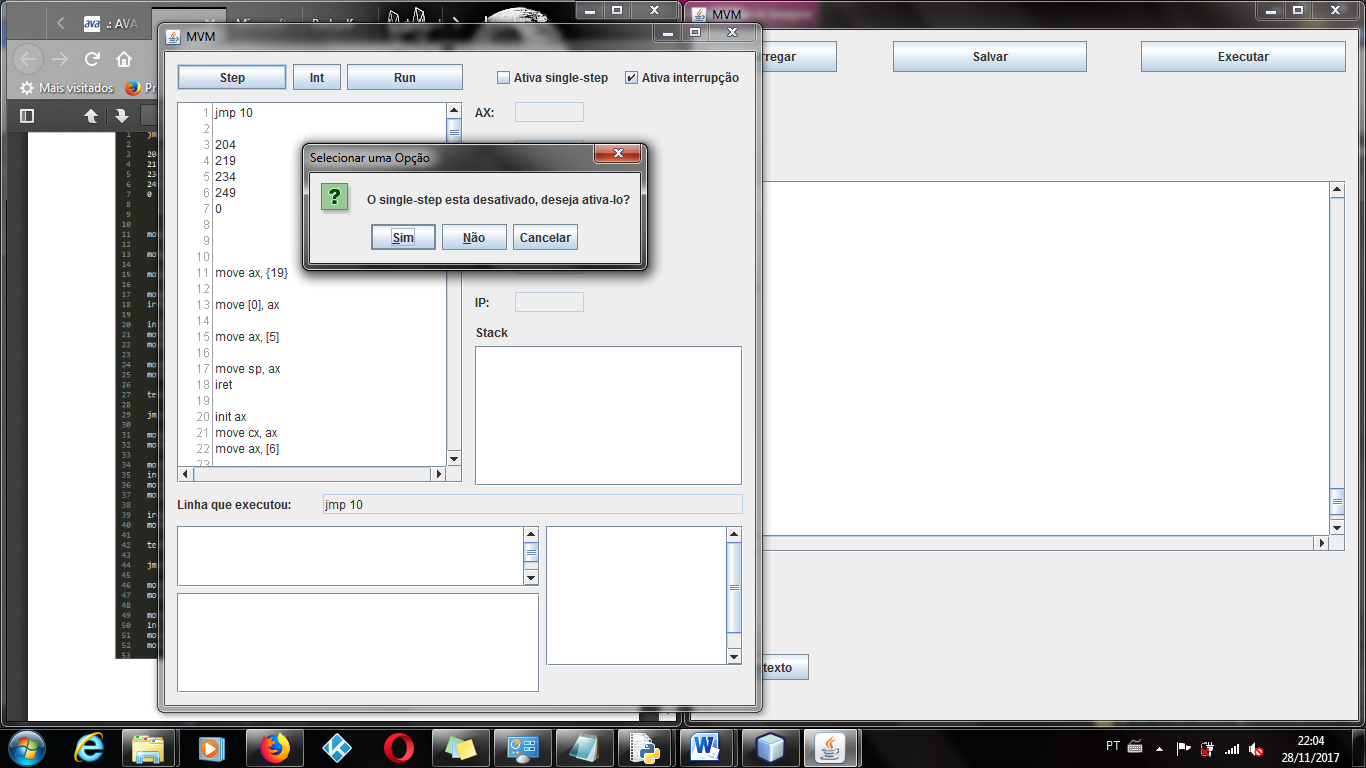


1. Adicionamos também o Controle de Ativação da Interrupção -TelaExecucao.java – Linha 451



TESTES

Single-Step



Interrupção

