

**Instruções:**

1. Esta atividade deve ser realizada em **Duplas**.
2. A defesa da implementação e a data final da postagem do código fonte comentado estarão devidamente indicadas no AVA2. Defesas/entregas feitas na aula seguinte terão 70% da nota.
3. A implementação poderá ser realizada em **qualquer** linguagem de programação, sendo que o código deverá ser entregue (também é válido o link para o trabalho no GitHub) e apresentado.
4. Todos os integrantes do grupo deverão estar presentes para apresentação e os códigos deverão conter identificação dos autores. Se algum integrante faltar, este terá que apresentar sozinho na semana seguinte e estará sujeito a *instrução 2*.
5. Se forem identificados trabalhos com grau de similaridade que caracterize cópia ou adaptação (autorizadas ou não pelos seus autores originais), a nota dos grupos será a nota de um trabalho dividida pelo número de grupos que entregou esses trabalhos similares.

## ATIVIDADE

---

Elaborar um programa que, mediante a entrada de um arquivo de memória de instrução (ROM) em linguagem de máquina (hexadecimal ou binário), o programa deverá identificar os **conflitos de dados e de controle** e implementar a solução para o problema.

1. Detectar a existência de conflito de dados.
  - a **[0,5 pontos]** Sem forwarding existente.
  - b **[0,5 pontos]** Com forwarding existente.
2. Detectar a existência de conflito de controle.
  - a **[0,5 pontos]** Sem forwarding existente.
  - b **[0,5 pontos]** Com forwarding existente.
3. Corrigir de conflito de dados com a inserção de NOPS.
  - a **[1,0 pontos]** Sem forwarding existente.
  - b **[1,0 pontos]** Com forwarding existente.
4. Corrigir conflito de controle com a inserção de NOPS.
  - a **[1,0 pontos]** Sem forwarding existente.
  - b **[1,0 pontos]** Com forwarding existente.
5. Implementar a solução integrada de dados e controle.
  - a **[1,0 pontos]** Sem forwarding existente.
  - b **[1,0 pontos]** Com forwarding existente.
6. **[2,0 pontos]** Recalcular os endereços após as correções de conflitos.

Demonstrar o sobrecusto de instruções da implementação de cada técnica. Ou seja, quantas instruções a mais tiveram que ser incluídas para resolver os conflitos.

## VALIDAÇÃO

---

Para a validação, utilize códigos assembly produzidos na disciplina de ARQ1. As soluções deverão ser demonstradas pela execução de um código disponibilizado . Será necessário ler os arquivos hexadecimais/binários manualmente usando o RARS (*dump file*).

## FLUXO DE OPERAÇÃO

---

Passos da operação:

1. Escolha o arquivo com o programa em binário ou **hexadecimal**;
2. Gere um arquivo novo com a aplicação de cada técnica; e
3. Exiba, no console, o sobrecusto de implementação de cada técnica implementada.

## CÁLCULO DA NOTA

---

As notas listadas na atividade só serão validadas mediante demonstração da execução correta além de domínio do algoritmo implementado.