

## **Cronograma da Disciplina Arquitetura de Computadores**

**Professor: Luís Fabrício W. Góes**

**Curso: Engenharia de Software – PUC Minas Praça da Liberdade – 2º Semestre de 2020**

<b>Data</b>	<b>Atividades</b>
04/08	Apresentação da Disciplina
11/08	História e Conceitos de Arquitetura de Computadores
18/08	Linguagens de Montagem: Aritméticas (4 pontos)
25/08	Linguagens de Montagem: Acesso à Memória (4 pontos)
01/09	Linguagens de Montagem: Desvio (IF-ELSE) (4 pontos)
08/09	Linguagens de Montagem: Desvio (SWITCH-CASE) (4 pontos)
15/09	Linguagens de Montagem: Repetição (WHILE) (4 pontos)
22/09	Linguagens de Montagem: Repetição (FOR) (4 pontos)
29/09	Linguagens de Montagem: Vetores (4 pontos)
06/10	Linguagens de Montagem: Chamadas de Sistema (4 pontos)
20/10	Linguagens de Montagem: Procedimentos Folha (4 pontos)
27/10	Sistemas Lógicos: Portas Lógicas (4 pontos)
03/11	Sistemas Lógicos: Circuitos Lógicos (4 pontos)
10/11	Caminho de Dados: Banco de Registradores e ULA
17/11	Caminho de Dados: Monociclo (4 pontos)
24/11	Pipeline: Adiantamento de Dados (4 pontos)
01/12	Memória Cache: Mapeamento Direto (4 pontos)
15/12	Memória Cache: Associativa por Conjunto (4 pontos)
22/12	Reavaliação (60 pontos)

### **Atividades Semanais (60 pontos) – Individual**

Toda semana deve ser postada uma solução de uma atividade no Canvas. Serão ao todo 15 atividades, cada uma valendo 4 pontos.

### **Trabalho Prático (40 pontos) – Em dupla**

O trabalho consiste em implementar uma rede neural perceptron de um neurônio em assembly utilizando um simulador QtSpim (<http://spimsimulator.sourceforge.net/>).

Um exemplo de código inicial e como executá-lo no QtSpim é apresentado no seguinte tutorial do Eitas! (<http://www.eitas.com.br/tutorial/10/56>).

A dupla deve criar um canal no Youtube (ou postar no canal de um dos integrantes do grupo) e postar o vídeo (com duração de 2 a 4 minutos) mostrando o uso do QtSpim, explicando a execução do código (mostrando os registradores, as instruções etc.)

Cada integrante da dupla deve também criar uma conta no Github e postar o código utilizado como exemplo no vídeo (arquivo com extensão .s), devidamente comentado.

A entrega do trabalho deve ser feita até o dia **15/12**. Deve ser postado no Canvas um texto com dois links: i) vídeo no youtube; ii) código no github.

**Reavaliação (60 pontos) –** A reavaliação é uma média simples entre o número total de pontos das suas atividades semanais e uma prova de 60 pontos.

## **Bibliografia**

PATTERSON, David A.; HENNESSY, John L. Organização e projeto de computadores: a interface hardware/software. Rio de Janeiro: Elsevier, Campus, c2014. xix, 709 p. ISBN 9788535235852

TANENBAUM, Andrew S.; AUSTIN, Todd. Organização estruturada de computadores. 6. ed. São Paulo, SP: Pearson Education do Brasil, 2013. xvii, 605 p. ISBN 9788581435398

WEBER, Raul Fernando. Fundamentos de arquiteturas de computadores. 4. ed. Porto Alegre: Bookman, 2012. xxiii, 400 p. ISBN 9788540701427

MCCOOL, Michael D; ROBISON, Arch D; REINDERS, James. Structured parallel programming: patterns for efficient computation. Amsterdam: Morgan Kaufmann, c2012. xxvi, 406 p. ISBN 9780124159938

MONTEIRO, Mário A. Introdução à organização de computadores. 5. ed. Rio de Janeiro: LTC - Livros Técnicos e Científicos, c2007. xii, 696 p. ISBN 9788521615439

PATTERSON, David A.; HENNESSY, John L. Arquitetura de computadores: uma abordagem quantitativa. Rio de Janeiro, RJ: Elsevier, Campus, c2014. xxv, 435, [274], p. ISBN 9788535261226

STALLINGS, William. Arquitetura e organização de computadores: projeto para o desempenho. São Paulo: Pearson, 2010. xiv, 624 p. ISBN 9788576055648

TOCCI, Ronald J.; WIDMER, Neal S.; MOSS, Gregory L. Sistemas digitais: princípios e aplicações. 11. ed. São Paulo: Pearson Prentice Hall, 2011. xx, 817, [2] p. ISBN 9788576059226