



<b>Disciplina:</b> Fundamentos da Computação		<b>Visto:</b>
<b>Professor:</b> Abrantes Araújo Silva Filho		
<b>Aluno:</b>		
<b>Turma:</b>	<b>Semestre:</b>	<b>Valor:</b> —
<b>Data:</b>	<b>Diário 0:</b> Introdução	

**Unidade 0: Introdução**  
— **Diário de Aprendizagem** —

- Este **Diário de Aprendizagem** é uma das atividades integrantes da disciplina de **Fundamentos da Computação** do curso de Ciência da Computação, Universidade Vila Velha (UVV).
- A confecção do diário de aprendizagem é atividade **obrigatória e altamente recomendada** por três motivos: a) você aprenderá muito mais a matéria se mantiver o diário; b) ao entregar todos os diários ao professor você está cumprindo parte das atividades avaliativas que contam pontos na disciplina (10% da nota); e c) as provas bimestrais discursivas seguirão o formato e conteúdo das perguntas do diário.
- Se você tiver dificuldade em responder alguma questão do diário, estude novamente a matéria. Se você realmente entendeu a matéria, não deveria ter muita dificuldade para responder o diário.
- Responda com caneta ou lápis escuro (2B, 4B, 6B).
- Verifique no calendário de sua turma a **data de entreg**. Após uma rápida avaliação e visto pelo professor ou pelos monitores, seu diário será devolvido.
- O diário não será corrigido pelo professor: cabe a você estudar e dar a resposta correta para todas as questões. Obviamente o professor está à disposição para esclarecimento de dúvidas, e os monitores podem auxiliar caso você tenha dificuldade.
- Manter o diário de aprendizagem atualizado pode ser a diferença entre você aprender a matéria e ser aprovado, ou não aprender a matéria e não ser aprovado.
- Bons estudos!

## 1 Por que você está aqui?

1. Pense um pouco sobre o motivo que te levou a fazer este curso e esta disciplina. Por que você está aqui?

---

---

---

2. Independentemente do motivo que o trouxe até aqui, se você está estudando ciência da computação você irá, de modo geral, aprender a fazer o quê?

---

---

3. Ciência da computação é aplicável somente à disciplinas na área de STEM (*science, technology, engineering, mathematics* — ciências, tecnologia, engenharia e matemática)? Por quê?

---

---

---

## 2 Computação não é sobre computadores ou programação

4. Um engano comum de quem começa na área da ciência da computação é confundir essa disciplina com a programação, ou achar que computação só pode ser feita através de computadores. Leia a [explicação de Hal Abelson<sup>1</sup>](#) que nos mostra que ciência da computação não se trata de computadores. Reflita sobre o que você leu e, depois, explique com suas próprias palavras o motivo porque consideramos que ciência da computação não é sobre computadores ou sobre programação.

---

---

---

---

---

---

---

---

---

---

---

5. Qual é a **essência** da computação?

---

---

---

<sup>1</sup><https://cmprz.me/cc-abelson>

6. Em relação à disciplina:

(a) Quais são os 4 focos principais? (um em cada linha)

---

---

---

---

(b) Cite 5 tópicos que serão abordados (um em cada linha)

---

---

---

---

---

### 3 O “arquipélago” da computação

7. Durante a aula o professor fez uma analogia da disciplina de Introdução à Ciência da Computação com um pequeno **barco visitando diversas ilhas de conteúdo**. Também foi dito que faremos uma exploração mais detalhada de algumas ilhas e, em outras, ficaremos apenas “na praia”. Explique porque essa analogia ilustra muito bem a disciplina.

---

---

---

---

---

8. Por que as ilhas que representam os fundamentos da computação e os fundamentos da programação são as maiores?

---

---

---

### 4 Não desista, você conseguirá!

9. Durante a aula o professor falou que você **beberá de uma mangueira de incêndio**. Explique o que isso quer dizer.

---

---

---

---

---

10. Cite e explique quais são as características e atitudes pessoais que promoverão seu sucesso na disciplina.

---

---

---

---

---

11. O que é o “momento Eureka” citado na aula? (assista o vídeo para lembrar)

---

---

---

12. Explique a frase: “Se você começar a resolver um problema colocando a mão no teclado para programar, você já está errando”.

---

---

---

## 5 Como será sua avaliação?

13. Ao final do semestre, como você saberá se foi bem na disciplina? Como você será avaliado?

---

---

---

14. Se você cumprir todas as atividades de programação da CR6.100B, que benefício você pode receber diretamente da Universidade de Harvard?

---

---

## 6 Construção e dinâmica da disciplina

15. Explique qual foi a base para a construção da CR6.100B, ou seja: qual (ou quais) foram as bases a partir das quais esta disciplina foi criada? Todas são igualmente importante?

---

---

---

---

16. Apesar de existirem linguagens de programação mais modernas, como Python, Go, Lua e outras, grande parte de nosso aprendizado será feito com uma linguagem de programação antiga, a **linguagem C**. Leia a explicação de David J. Malan, professor da disciplina de introdução à ciência da computação de Harvard (em <https://cmprz.me/malan-c>) e explique, em poucas palavras, porque utilizaremos essa linguagem.

Obs.: Adicionalmente, se você quiser complementar sua pesquisa, pode visitar o YouTube e ouvir o podcast *Why C* (em <https://cmprz.me/malan-c-podcast>), onde David Malan dá uma explicação mais detalhada.

---

---

---

---

---

---

---

---

---

---

---

---

17. Visite o site *Rubber Duck Debugging* (em <https://cmprz.me/ddb>) e explique porque você precisará de um patinho de borracha para nossa disciplina.

---

---

---

---

---

---

---

---

## 7 Unidades de estudo

18. Se a última unidade de estudo é a número 17, podemos afirmar que temos exatamente 17 unidades de estudo no total. Verdadeiro ou falso? Por quê?

---

---

19. Considere a seguinte afirmação: “se eu tenho 6 elementos ( $n = 6$ ), então esses elementos devem ser identificados pelos números de 1 até 6 (de 1 até  $n$ ), e não pelos números de 0 até 5 (de 0 até  $n - 1$ )”. Você concorda com essa afirmação? Por quê?

---

---

---

---

---

---

---

---

20. Qual o objetivo do **projeto final** da disciplina?

---

---

---

21. Em qual unidade faremos um estudo teórico sobre a computação?

---

---

22. Quais unidades da CR6.100B não existem na CS50 original?

---

---

---

23. Esta disciplina alcançará todas as áreas de conhecimento da computação? Esta disciplina irá prepará-lo para estudar por conta própria no futuro?

---

---

---

24. Que linguagem de programação é utilizada para trabalhar com sistemas de gerenciamento de bancos de dados?

---

25. Quem foi o criador da linguagem Python?

---

26. Ao estudar sobre programação para a Internet, você aprenderá a utilizar três linguagens especiais. Quais são elas?

---

## 8 *Blown to Bits*

27. Explique, em poucas palavras, o que você entendeu e aprendeu com a leitura das páginas 1–4 do *Blown to Bits*.

---

---

---

---

---

---

---

---

---

---

## 9 Desafios de conhecimento

28. Pesquise sobre **Edsger Wybe Dijkstra**, um cientista da computação holandês que foi fundamental no desenvolvimento de diversos fundamentos da computação (inclusive criando diversos algoritmos importantíssimos). Escreva um pequeno texto descrevendo as principais contribuições de Dijkstra para a computação e para a ciência da computação.

---

---

---

---

---

---

---

---

29. Todos os trabalhos publicados (e alguns manuscritos não publicados) de Dijkstra<sup>2</sup> estão disponíveis para consulta e download no “[E. W. Dijkstra Archive](#)”<sup>3</sup>. O manuscrito “831: *Why numbering should start at zero*”<sup>4</sup> é interessante e explica o motivo do cientista da computação começar a contar do zero. Apesar de ser um manuscrito pequeno (apenas três páginas), sua leitura não é muito fácil. Tente ler esse texto e faça o melhor possível para explicar porque o cientista da computação deve começar a contar no zero.

Obs.: não se preocupe se você não entender o texto completamente... é sua primeira experiência de leitura de um texto real de ciência da computação, e é natural que você não consiga entender. Com o tempo você aprenderá!

---

---

---

---

---

---

---

---

30. A síndrome do encarceramento (*locked-in syndrome*) é uma condição onde uma pessoa está totalmente paralisada por causa de um derrame. Você está totalmente consciente, com sua inteligência totalmente preservada, pode ouvir, pensar, ver... mas não pode falar. Infelizmente não existe cura. Esse é um problema para o qual a ciência da computação pode contribuir! Pense um pouco e responda:

---

<sup>2</sup>Você ouvirá falar de Dijkstra diversas vezes ao longo do curso de ciência da computação!

<sup>3</sup><https://cmprz.me/dijkstra-arch>

<sup>4</sup><https://cmprz.me/dijkstra-zero>

- (a) Pesquise sobre **Jean-Dominique Bauby**, que foi um portador da síndrome de encarceramento, e tente descobrir como ele foi capaz de, sem tecnologia nenhuma, descobrir como se comunicar e até mesmo escrever um livro: “O Escafrando e a Borboleta”, sua autobiografia que narra sua vida após acordar no hospital, totalmente paralisado.

---

---

---

---

---

---

---

---

---

---

---

---

---

---

- (b) O que você, como um cientista da computação, poderia fazer para ajudar pessoas com a síndrome de encarceramento voltarem a “falar”?

---

---

---

---

---

---

---

---

---

---

---

---

---