

# FUNDAMENTOS DE COMPUTAÇÃO E ALGORITMOS

---

Mário Souto e Edson Ifarraguirre Moreno

“

Quando você está na computação, pensa muito em problema e solução, mas quando a gente vai em uma escola de arte, os artistas expressam uma paixão com algo que eles querem expressar.

”

*Katia Canepe Vega*

# Conheça o livro da disciplina

## CONHEÇA SEUS PROFESSORES 3

*Conheça os professores da disciplina.*

## EMENTA DA DISCIPLINA 4

*Veja a descrição da ementa da disciplina.*

## BIBLIOGRAFIA DA DISCIPLINA 5

*Veja as referências principais de leitura da disciplina.*

## O QUE COMPÕE O MAPA DA AULA? 6

*Confira como funciona o mapa da aula.*

## MAPA DA AULA 7

*Links de artigos científicos, informativos e vídeos sugeridos.*

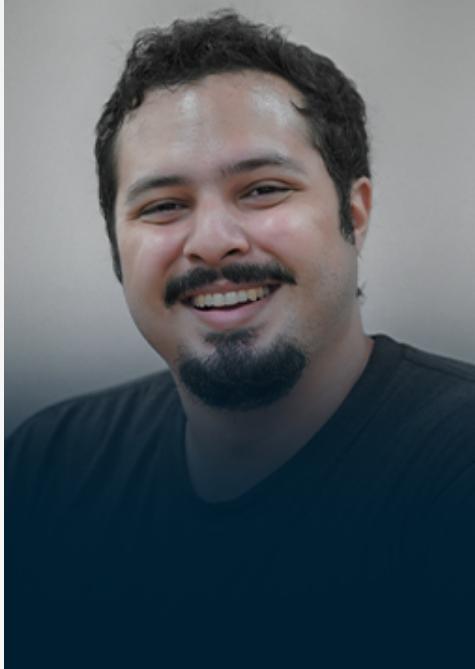
## RESUMO DA DISCIPLINA 33

*Relembre os principais conceitos da disciplina.*

## AVALIAÇÃO 34

*Veja as informações sobre o teste da disciplina.*

# Conheça seus professores



## MÁRIO SOUTO

Professor Convidado

Especialista em desenvolvimento web, começou a programar com 11 anos e não parou mais! Atualmente é Engenheiro de Software no Nubank, GitHub Star, Microsoft MVP, Alura Star e criador e diretor do Canal DevSoutinho no YouTube, onde lança vídeos semanalmente com foco no mundo da tecnologia e programação.

## EDSON IFARRAGUIRRE MORENO

Professor PUCRS

Edson Ifarraguirre Moreno é doutor em Ciência da Computação pela Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS). Atualmente é professor adjunto pela mesma PUCRS, estando vinculado a Escola Politécnica da pucrs, sendo responsável por disciplinas da área de hardware para os cursos de ciência da computação, engenharia da computação. Adicionalmente trabalha com a orientação de alunos no desenvolvimento de projetos do curso de Engenharia de Software. Desde 2016 coordena o laboratório iSeed Labs, uma parceria entre a academia e a iniciativa privada que tem por objetivo fomentar a inovação e o empreendedorismo. Os principais temas de pesquisa incluem: Sistemas multiprocessados em chip (em inglês, Multiprocessor System on chip, MPSoC), projeto em nível de sistema e redes em chip (em inglês, Network on chip, NoC).



# *Ementa da Disciplina*

Entendimento de algoritmos e estruturas de dados (listas, filas, pilhas e árvores) na solução de problemas. Análise de algorítmica quanto a aplicação e complexidade. Análise da aplicabilidade e manejo de estruturas de dados lineares e hierárquicas.

# *Bibliografia da Disciplina*

As publicações destacadas têm acesso gratuito.

## **Bibliografia básica**

**GRONER, Loiane. Learning JavaScript Data Structures and Algorithms. Third Edition. Birmingham: Packt, 2018.**

CORMEN, Thomas H.; et al. Algoritmos: teoria e prática. Terceira Edição. Rio de Janeiro : Elsevier, 2012.

PIVA JR, Dilermando et al. Estrutura de dados e técnicas de programação. 2a. Edição. Rio de Janeiro: Elsevier, 2019.

## **Bibliografia complementar**

FLANANGAN, David. Java Script: the definitive Guide. Seventh Edition. O'Reilly, 2020.

GONÇALVES, Marcela dos Santos. Algoritmos e programação. Porto Alegre: Sagah , 2018.

PIVA JR, Dilermando et al. Algoritmos e Programação de Computadores. 2a. Edição. Rio de Janeiro : Elsevier, 2019.

**TOSCANI, Laira Vieira; VELOSO, Paulo. Complexidade de Algoritmos. Porto Alegre: Bookman, 2012.**

DASGUPTA, Sanjoy; PAPADIMITRIOU, Christos; VAZIRANI, Umesh. Algoritmos. Porto Alegre: AMGH, 2010.

# O que compõe o Mapa da Aula?

## MAPA DA AULA

São os capítulos da aula, demarcam momentos importantes da disciplina, servindo como o norte para o seu aprendizado.



## EXERCÍCIOS DE FIXAÇÃO

Questões objetivas que buscam reforçar pontos centrais da disciplina, aproximando você do conteúdo de forma prática e exercitando a reflexão sobre os temas discutidos. Na versão online, você pode clicar nas alternativas.



## PALAVRAS-CHAVE

Conceituação de termos técnicos, expressões, siglas e palavras específicas do campo da disciplina citados durante a videoaula.



## VÍDEOS

Assista novamente aos conteúdos expostos pelos professores em vídeo. Aqui você também poderá encontrar vídeos mencionados em sala de aula.



## PERSONALIDADES

Apresentação de figuras públicas e profissionais de referência mencionados pelo(a) professor(a).



## LEITURAS INDICADAS

A jornada de aprendizagem não termina ao fim de uma disciplina. Ela segue até onde a sua curiosidade alcança. Aqui você encontra uma lista de indicações de leitura. São artigos e livros sobre temas abordados em aula.



## FUNDAMENTOS

Conteúdos essenciais sem os quais você pode ter dificuldade em compreender a matéria. Especialmente importante para alunos de outras áreas, ou que precisam relembrar assuntos e conceitos. Se você estiver por dentro dos conceitos básicos dessa disciplina, pode tranquilamente pular os fundamentos.

## CURIOSIDADES

Fatos e informações que dizem respeito a conteúdos da disciplina.

## DESTAQUES

Frases dos professores que resumem sua visão sobre um assunto ou situação.

## ENTRETENIMENTO

Inserções de conteúdos para tornar a sua experiência mais agradável e significar o conhecimento da aula.

## CASE

Neste item, você relembra o case analisado em aula pelo professor.

## MOMENTO DINÂMICA

Aqui você encontra a descrição detalhada da dinâmica realizada pelo professor.

# Mapa da Aula

Os tempos marcam os principais momentos das videoaulas.

## AULA 1 • PARTE 1

02:57



**“** Software é o que você xinga e hardware é o que você chuta. **”**

### FUNDAMENTO

#### Software e Hardware

Software é um sistema ou programa de computador, tecnicamente, trata-se de uma sequência de instruções a serem seguidas e/ou executadas, na manipulação, redirecionamento ou modificação de um dado (informação) ou acontecimento. Ou seja, inclui não só o programa de computador, propriamente dito, mas, também, manuais e especificações.

04:31



### PALAVRAS-CHAVE

**Hacker typer:** Um site criado com intuito cômico de tornar seus “pseudo sonhos hacker” uma realidade.

Acesse o site em: <https://hackertyper.net>

O Hardware é a parte física de um computador ou outros dispositivos, o conceito se aplica à unidade central de processamento, à memória e aos dispositivos de entrada e saída, bem como a todos os componentes físicos de computadores. As demais ferramentas como mouse e teclado, por exemplo, são chamados de periféricos.

05:00



### As grandes empresas

FANG (ou MANG) é uma sigla popular no mundo da tecnologia, e representa as cinco empresas que mais pagam bem no mercado de tecnologia atualmente, são elas: Facebook, Amazon, Apple, Netflix e Google. Todas essas empresas têm alguma ligação entre si, além de terem estruturas gigantescas.

Existe um grande investimento externo na Amazon, especificamente, devido a sua estrutura de servidores, inicialmente desenvolvida para suportar os acessos a seu e-commerce durante feriados como a Black Friday, passou a ser vendida como serviço e é uma das maiores fontes de lucro da empresa no momento.

### PALAVRA-CHAVE

**Vale do Silício:** Localizado na Califórnia, é um dos maiores polos tecnológicos do mundo, e abriga muitas start-ups e empresas globais de tecnologia como: Apple, Facebook e Google.

## PALAVRAS-CHAVE

**Latência:** Considerada uma medida de velocidade da conexão de internet, é a diferença de tempo entre o início de um evento e o momento em que os seus efeitos se tornam perceptíveis.

## História

ENIAC foi o primeiro computador digital eletrônico de grande escala, suas funções não passavam muito além de uma calculadora, algo que hoje é básico para aparelhos muito menores como um smartphone. Hoje, o chip mais recente desenvolvido pela Apple possui uma capacidade enormemente maior do que o primeiro computador, e o mercado faz questão de mostrar isso em números para deixar claro o que é melhor ou não.

Porém, números sozinhos não mostram nada, é preciso uma base de tecnologia para uma decisão de compra. Quando falamos de tecnologia, o pensamento chave é entender qual o problema que se quer resolver, uma vez que se entende aonde se quer chegar é possível trabalhar um problema para entender qual é o ponto inicial que se pode atacar e evoluir em cima disso.

06:43



10:10



## PALAVRA-CHAVE

**Servidor em nuvem:** É um grupo de máquinas físicas trabalhando juntas, como uma só, onde caso ocorra algum problema com uma delas, outra máquina assumirá a carga, impedindo que um sistema ou site fique fora do ar.

11:03



11:14



*“A base vai muito de se questionar como as coisas funcionam.”*



## EXERCÍCIO DE FIXAÇÃO

Segundo o professor Mário Souto, o que faz a tecnologia avançar?

16:36



## PALAVRA-CHAVE

**MVP:** Mínimo produto viável, em empreendedorismo, é a versão mais simples de um produto que pode ser lançada com uma quantidade mínima de esforço e desenvolvimento.

## Como a internet funciona?

Os sites que visitamos na internet estão armazenados em diversos computadores/servidores espalhados pelo mundo, e funciona na seguinte estrutura: um cliente que acessa um servidor solicitando e recebendo dados em resposta.

No contexto web temos sempre o cliente, o servidor e o banco de dados, onde o servidor faz o papel tanto de receber uma requisição quanto de fazer uma requisição, a um banco de dados. Neste modelo, sempre temos a entrada, o processamento e a saída, sendo que o profissional de tecnologia, em geral, trabalha na parte de processo.



18:35



11:03



*Você trabalhando no dia a dia, tem que ter clareza não só do que você está vendo como tecnologia, mas do lugar que você trabalha.*



25:08

## PALAVRA-CHAVE

**Front-end:** É o desenvolvimento da interface gráfica do usuário de um site, por meio do uso de HTML, CSS e JavaScript, para que os usuários possam visualizar e interagir com os sites.



25:38



27:42

## Conexão cliente-servidor

Algo muito comum na internet é a “queda” de um servidor devido a quantidade de acessos, isso ocorre porque o sistema de request e response entre o cliente e o servidor, é uma atividade síncrona, ou seja, é necessário que o servidor esteja disponível para atender ao request e comunicar com o cliente.

O servidor confirma sua disponibilidade, então recebe a requisição, então mantém conectado ao cliente até enviar todos os dados solicitados para acesso ao site, encerrando então essa ligação. Quando há uma fila de clientes muito grande para o servidor atender, isso pode exceder o limite de processamento de requisições do servidor, fazendo com que ele trave e pare de funcionar.

Um fator que contribui para esse problema, são as novas tentativas de conexão por parte do usuário, visto que o servidor começa a acumular chamadas não atendidas e os dados de cada chamada, causando o erro 500, que é o erro interno do servidor, informando que não consegue mais processar.

## PALAVRAS-CHAVE

**Backend:** É a estrutura que possibilita a operação do sistema, a criação do código e manutenção de um site ou sistema, incluindo bancos de dados, servidores e aplicativos.

**“** Na humanidade, a gente tem vários processos que a tecnologia, por mais que pareça que ela cria coisas novas, ela absorve muitas coisas de outros conhecimentos que a gente tem. **”**



30:33

## Como resolver?

Uma forma de resolver o problema de muitas pessoas acessando um sistema ao mesmo tempo, é a utilização de dois sistemas, porém isso também exige alguns cuidados devido à maior complexidade. Outra forma é a técnica load balancer que, quando um sistema é sobrecarregado, ele passa uma parte dessa carga para outro sistema, fazendo o balanço.

Todas as técnicas terão sempre seu ônus e seu bônus, por isso as soluções precisam ser alinhadas com o restante dos setores da empresa, a fim de achar a melhor solução.



35:42



## EXERCÍCIO DE FIXAÇÃO

Qual a técnica que faz um balanço de carga entre sistemas?

**“** Quanto mais automático, mais você paga para alguém fazer, quanto menos automático mais o time da sua empresa vai ter que fazer. **”**

## PALAVRAS-CHAVE

**Fred Brooks:** É um engenheiro de software e informático estadunidense, conhecido pelo gerenciamento do projeto do OS/360 e por suas publicações.



36:54



38:26

## PALAVRA-CHAVE

**12factors:** A metodologia doze-fatores pode ser aplicada a aplicações escritas em qualquer linguagem de programação, e que utilizem qualquer combinação de serviços de suportes (banco de dados, filas, cache de memória, etc).

Saiba mais em: [https://12factor.net/pt\\_br/](https://12factor.net/pt_br/)



38:56



45:00

## LEITURA INDICADA

### No silver bullet

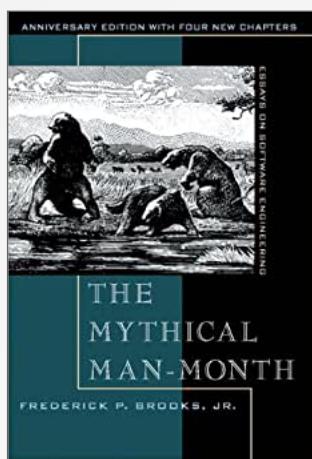
Artigo amplamente discutido sobre engenharia de software, onde o autor argumenta que “não existe um desenvolvimento único, seja em tecnologia ou técnica de gerenciamento, que, por si só, promete uma melhoria de uma ordem de magnitude [dez vezes] em uma década em produtividade, confiabilidade e simplicidade.”

## LEITURA INDICADA

Livro: The Mythical Man-Month



39:53



É um livro de gerenciamento de projeto de software cujo tema central é “adicionar força de trabalho para um projeto de desenvolvimento de software já atrasado, atrasa-o ainda mais”, ideia conhecida como Lei de Brooks.

## AULA 1 • PARTE 2

### Escalonamento

Escalar é um processo difícil de tornar um servidor com poucos acessos ser apto a receber muitos acessos, ou seja, aumentar a sua complexidade. Existem duas opções para escalar:

Verticalmente: uma máquina maior que processa tudo, exige mais recurso;

Horizontalmente: várias máquinas menores, exige mais tempo. O load balancer é um exemplo de escalonamento horizontal.

Os servidores possuem conteúdos estático, server side rendered e banco de dados. Conteúdo estático é aquele que não muda como, por exemplo, uma foto publicada em uma rede social. O server side rendered é o processo de renderizar conteúdo dinâmico para estático.

00:08



01:23



### PALAVRA-CHAVE

**Infoq:** Site que busca ajudar times de desenvolvimento a adotarem novas práticas e tecnologias. Seu conteúdo foi redigido por pessoas desenvolvedoras.

Saiba mais em: <https://www.infoq.com.br>

## EXERCÍCIO DE FIXAÇÃO

Quais os 3 tipos de conteúdo dos servidores?

**Cada sistema vai desempenhar de uma forma baseado no histórico dele.**

### PALAVRAS-CHAVE

**Server side rendered:** É uma técnica usada no desenvolvimento da Web que produz uma resposta personalizada para a solicitação de cada usuário (cliente) ao site. A alternativa é o próprio servidor web entregar uma página web estática.



03:44



05:20

### PALAVRAS-CHAVE

**Cloudflare:** Empresa que fornece uma rede de distribuição de conteúdo, serviços de segurança da Internet e serviços de servidor de nome de domínio distribuído, localizados entre o visitante e o provedor de host do usuário da Cloudflare, agindo como um proxy reverso para sites.



06:04



06:21

### PALAVRA-CHAVE

**Cache:** Tem o objetivo de evitar o acesso ao dispositivo de armazenamento, armazenando os dados em meios de acesso mais rápidos. Seu uso visa obter uma velocidade de acesso a memória próxima da velocidade de memórias mais rápidas, e ao mesmo tempo disponibilizar no sistema uma memória de grande capacidade.

### PALAVRAS-CHAVE

**Amazon EC2:** É uma parte central da plataforma de cloud computing da Amazon, que permite que os usuários aluguem computadores virtuais nos quais rodam suas próprias aplicações.



07:56



08:39

### PALAVRA-CHAVE

**Vercel:** Plataforma voltada para a hospedagem de aplicações, conhecida por ser a empresa criadora do framework Next JS, voltado para o React. Também é possível realizar o deploy de aplicações Nuxt na plataforma.

## Fila e performance

O conceito de fila vem do mundo físico para o digital, e se torna um recurso muito eficiente para evitar problemas de sobrecarga de processamento de um servidor. A fila vai gerar uma senha para cada cliente que quer acessar o servidor fazendo com que esses sejam atendidos um a um, ou na capacidade suportada pelo servidor, e em ordem de chegada. A fila é uma estrutura de dados que representa a capacidade de fazer algo em uma ordem, sendo uma das soluções para o problema de acessos simultâneos.

Outro problema muito comum é a dificuldade de acesso, quando um sistema não é muito bem otimizado, dependendo da velocidade de internet do usuário, o acesso pode ser dificultado. Do ponto de vista dos negócios, isso gera prejuízos financeiros, uma vez que os usuários não conseguem acessar o site para consumir o produto ou serviço, sendo direta ou indiretamente.

## Espera Passiva e Ativa

Como visto, entendemos que performance é experiência de usuário (UX), o usuário precisa ter a sensação de que o site é rápido para permanecer utilizando o serviço. Existem dois conceitos importantes em performance, que são a espera passiva, onde o tempo aparenta demorar mais para passar, e a espera ativa, onde se apresenta uma distração para o usuário, fazendo com que o tempo aparente passar mais rápido.

Uma técnica utilizada na espera ativa é minificar e concatenar arquivos, removendo aquilo que for irrelevante nos arquivos do site. Outra forma é trazer o conteúdo para mais perto do usuário, diminuindo a latência.

Ao trabalhar com performance sempre existirá um ponto que ao tentar corrigir, vai causar impacto em outro local.



10:43



12:34

## PALAVRA-CHAVE

**Trade-off:** Define uma situação em que há conflito de escolha, se caracteriza em uma ação econômica que visa à resolução de problema mas acarreta outro, obrigando uma escolha.



13:52

*“ Praticamente tudo que a gente tem em tecnologia, dá para fazer um paralelo com algo do mundo real. ”*



16:08



19:16

## PALAVRA-CHAVE

**http archive:** O HTTP Archive Rastreia como a web é construída, rastreando periodicamente os principais sites da web e registrando informações detalhadas sobre recursos buscados, APIs e recursos da plataforma da Web usados e rastreamentos de execução de cada página.

Saiba mais em: <https://httparchive.org/>

## VÍDEO

### Perceived Performance



Palestra de Fernanda Bernardo, que explica o que é provider performance, um termo da engenharia da computação que se refere a quanto rápida uma funcionalidade de software aparenta performar sua tarefa.

<https://www.youtube.com/watch?v=VqpnP4UewzU>



24:03

## VÍDEO

### Além dos milissegundos



Live de João Cunha, Product Manager da Nubank, onde ele explica como nossos cérebros entendem o tempo.

Traz, também, conceitos para aumentar a percepção de performance com técnicas que ele aplicou.

<https://www.youtube.com/watch?v=QZKjn0cm2Rw>



24:51

27:19

## LEITURA INDICADA

### Por uma Web mais rápida: 26 técnicas de otimização de Sites

A primeira versão desse post surgiu de uma palestra no QCon sobre otimização de performance de carregamento de páginas na Web. Desde então, ele virou o post mais acessado aqui no blog e gerou dezenas de outros artigos.

Saiba mais em: <https://www.alura.com.br/artigos/por-uma-web-mais-rapida-26-tecnicas-de-otimizacao-de-sites>

## PALAVRAS-CHAVE

**GTmetrix:** O GTmetrix foi desenvolvido pela Carbon60 como uma ferramenta para os clientes testarem facilmente o desempenho de suas páginas da web.

Acesse em: <https://gtmetrix.com/>



06:04

## PALAVRAS-CHAVE

**X86:** Família de processadores baseados no Intel 8086, da Intel Corporation. A arquitetura é chamada x86 porque os primeiros processadores desta família eram identificados somente por números terminados com a sequência “86”: o 8086, o 80186, o 80286, o 80386 e o 80486.

28:35



### Computador lento

No contexto de programação, é prática diária buildar projetos, compilar, testar algum código no navegador, e isso é feito por um processador. Outro componente importante para quem trabalha com programação é a quantidade de armazenamento, neste quesito existem as seguintes opções:

HDD: dispositivo de armazenamento que usa pratos mecânicos e um cabeçote móvel de leitura/gravação para acessar dados;

SSD: dispositivo mais rápido que armazena dados em chips de memória que podem ser acessados imediatamente;

M.2 NVME SSD: um SSD com otimização do caminho que o computador leva entre o comando e o acesso ou escrita do arquivo.

Para quem trabalha com mídia audiovisual, além dos componentes já mencionados, é importante uma placa de vídeo.

Outros componentes são a RAM, responsável por dar mais agilidade e velocidade no funcionamento geral do sistema, e a placa mãe, a principal placa de circuito impresso de um computador, que contém a unidade central de processamento.

## AULA 1 • PARTE 3

### Princípio de Programação

O navegador é um ambiente que possibilita o teste de códigos, segundo o professor Mário Souto ele é uma das melhores formas de programar, pois corta barreiras. Por meio do inspect, no navegador, é possível alterar informações da página e visualizar, auxiliando no processo de aprendizagem.

O professor apresenta alguns exemplos de como programar e realizar alguns testes por meio do navegador. E destaca que programar é dar uma instrução para o computador, sendo esse capaz de executar a instrução passada e a partir disso gerar algum resultado na máquina.

00:45



00:28



**“** Qualquer coisa que você faz no computador é código rodando. **”**



## PALAVRA-CHAVE

03:54 14:20



### Ferramenta de programação

Mesmo que seja possível utilizar o navegador para testes e aprendizado, quando de fato vamos programar utilizamos softwares como o Visual Studio Code. O professor Mário Souto comenta a base dessa ferramenta, e como iniciar um projeto de código apresentando ainda um exemplo básico de código. Destaca que na programação, o tipo da extensão do arquivo é muito importante, por exemplo, o navegador não abre arquivos txt.

A linguagem HTML, por padrão, interpreta tudo como texto, cada parte do código deve ter sua função especificada, pois o navegador é preparado para identificar algumas tags e executar ações específicas quando identifica essas tags.

**Debug:** Depuração, em português, é o processo de encontrar e reduzir defeitos num aplicativo de software ou mesmo em hardware.

00:45



“ A web foi feita para fazer documentos.”

27:52



### Refinando as perguntas

Entendendo a base da programação, o passo seguinte é refinhar as perguntas para solução de problemas. Lembre-se que a pergunta inicial sempre é muito superficial.

O professor Mário Souto apresenta um exemplo prático de código um pouco mais complexo a fim de demonstrar o conceito.

30:26



“ Dificilmente, um código fica pronto, sempre tem algum ponto que dá para melhorar.”

35:09



35:41



### Praticando

O professor Mário Souto destaca que refazer o que já existe é uma boa forma de ter um caminho para pensar no problema que está sendo resolvido. Utilizando a calculadora de preço da viagem da empresa Uber, demonstra como utilizar algumas tags básicas de código.

## PALAVRAS-CHAVE

**Github copilot:** Ferramenta de inteligência artificial desenvolvida pelo GitHub em conjunto com a OpenAI, afim de auxiliar usuários de ambientes de desenvolvimento integrados como Visual Studio Code, Visual Studio, Neovim e JetBrains por meio do sugestão de código para preenchimento automático.

## AULA 1 • PARTE 4

### Praticando: continuação

O professor Mário Souto segue com a demonstração prática destacando a importância da comunicação entre o setor de tecnologia e o de negócios. Lembra também a importância de fazer as perguntas certas para entender o problema e executar a melhor solução. O professor demonstra ainda como debugar por meio do inspect do navegador.



02:45



03:20

### PALAVRA-CHAVE

**Stakeholder:** Termo utilizado que se refere às partes interessadas que devem estar de acordo com as práticas executadas pela empresa.

### PALAVRAS-CHAVE

**Indentação:** Termo aplicado ao código fonte de um programa para ressaltar ou definir a estrutura do algoritmo. Empregada com o objetivo de ressaltar a estrutura do algoritmo, aumentando a legibilidade do código.



11:28

### EXERCÍCIO DE FIXAÇÃO

Qual a diferença entre a função e a variável visualmente no código?

### Praticando: continuação

O professor Mário Souto dá algumas dicas de programação:

Quando um código funciona, copie e cole ele no momento exato, ou reescreve com atenção. Muitos erros no início ocorrem por falta de atenção, costume com a estrutura, etc;

Quando o código começa a ficar muito extenso na horizontal, é indicado rever a organização desse.



14:50



15:52

 Trabalhar com dinheiro é todo um outro problema no mundo da tecnologia. 

## Praticando: continuação

O código demonstrado pelo professor Mário Souto apresentou a criação de variável, formatação de calor, criação de função, retorno de função e parâmetros, tópicos base para qualquer linguagem de programação.

Traz que a validação na entrada evita um problema muito comum na programação chamado de null pointer, um valor salvo para indicar que o ponteiro ou referência não se refere a um objeto válido. A validação na entrada garante uma maior segurança no código.

## Praticando: continuação

O professor Mário Souto segue com seu exemplo, apresentando um tipo de variável muito utilizada na programação é a booleana, que possui apenas dois valores possíveis: true(1) e false (0). Demonstra também que para encerrar a execução em uma função, utilizamos o return, que também é uma forma de economizar custo de processamento e uma boa prática.

Usar serviços de terceiros, como o Google, é chamado de API, um código ou conjunto de rotinas e padrões estabelecidos por um software para a utilização das suas funcionalidades por outros aplicativos.

## PALAVRAS-CHAVE

### Banco de dados de chave-valor:

Paradigma de armazenamento de dados projetado para armazenar, recuperar e gerenciar matrizes associativas e uma estrutura de dados mais comumente conhecida hoje como dicionário ou tabela hash.



23:08



24:00

“ Não tenha medo de errar,  
ou de fazer algo. ”



25:01

## LEITURA INDICADA

### Livro: Orientação a Objetos e SOLID para Ninjas



Livro de Mauricio Aniche que traz os conceitos básicos de Orientação a Objetos. Um paradigma que, além de fundamental, é utilizado em todo o mercado de software ao redor do mundo.



11:28



## EXERCÍCIO DE FIXAÇÃO

Como são chamados os valores true e false?

## AULA 2 • PARTE 1

“ Sistema, na verdade, é uma forma de organizar.”

### Inversão de controle e abstração

A inversão de controle é um princípio onde a sequência de chamadas dos métodos é invertida em relação à programação tradicional, ou seja, ela não é determinada diretamente pelo programador.

Ao aplicar a inversão de controle, é possível perceber que a abstração é o nome que damos para a função que formata valor. Ou seja, é a forma como se adequa o jeito que se programa para atender uma necessidade que uma empresa tem.

Existem diversas formas de resolver um mesmo problema, por isso é importante ter a visão do problema e os recursos disponíveis para pensar na melhor maneira de solucionar uma determinada necessidade. Normalmente, essa solução é apresentada por um time que analisa o problema, unindo seus conhecimentos para buscar a melhor forma de resolver.

00:36



### Trabalhando com banco de dados

Bancos de dados é onde a informação é armazenada, neles sempre teremos tabelas internas. A relação entre tabelas, ou vinculação de elementos, é o que chamamos de abstração, uma forma de reduzir a complexidade e tornar o projeto e a implementação mais eficientes em sistemas complexos.

03:05



10:23



### PALAVRA-CHAVE

13:53



**Datomic:** É um banco de dados distribuído e implementação do Datalog, tem transações ACID, junções e uma linguagem de consulta lógica, Datalog. Uma característica distintiva é que seu tempo é uma característica básica das entidades de dados.

Acesse em: <https://www.datomic.com/>

14:21



“ Um mesmo problema pode ser resolvido de formas diferentes, e a forma que você abstraia a informação vai depender do problema que te deram.”

15:22



“ Muitas vezes, achamos que mudar a ferramenta é o que resolve o problema, mas vai de você aprofundar o quanto você conhece da ferramenta que você está.”

## FUNDAMENTO

### Banco de dados relacional

Os BDR surgiram em meados da década de 1970, mas somente utilizados pelas empresas alguns anos mais tarde no lugar de arquivos simples, bancos de dados hierárquicos e em rede.

São bancos de dados que modelam os dados de forma que sejam percebidos pelo usuário como tabelas, ou mais formalmente relações.

Foram desenvolvidos para prover acesso facilitado aos dados, possibilitando que os usuários utilizassem uma grande variedade de abordagens no tratamento das informações.



17:27



18:54



Nenhum conhecimento que você aprende é desperdiçado.



19:34

### Wordpress

O Wordpress é uma ferramenta de blog, que utiliza um banco MySQL (banco de dados relacional), e que possui três tabelas: posts, categorias e páginas. Com essas três tabelas ele pode ser utilizado como uma ferramenta que abstrai a criação do banco de dados, e trabalhando bem com o wordpress é possível ter uma ferramenta fácil de prototipagem ou criação de sistemas.

### Navegador e estruturas de dados

É possível aprender muito da base de programação por meio do navegador, pois esse software que utilizamos muito para abrir sites, vem evoluindo muito para acompanhar e dar suporte aos hardwares que cada vez estão melhores.

A função do browser é ler HTML, CSS e JavaScript, além de lidar com requisição, ou seja, ele precisa saber o tipo de conteúdo que está sendo entregue e as informações dessa entrega. Outra parte do navegador é o histórico, que guarda as informações de acesso aos sites, o comportamento do usuário.

Além disso, também possui os favoritos, que tem uma função parecida com o histórico mas pode armazenar os sites em pastas.



23:10



## EXERCÍCIO DE FIXAÇÃO

São formatos lidos pelo navegador EXCETO:



23:40

### PALAVRA-CHAVE

**Adobe Flash Player:** Foi um reprodutor de multimídia e aplicações amplamente distribuído, anteriormente criado e disponibilizado pela Macromedia, que agora pertence à Adobe Systems.

## PALAVRAS-CHAVE

**HTML5:** Linguagem de marcação para a World Wide Web e é uma tecnologia chave da Internet, originalmente proposto por Opera Software, sendo a quinta versão da linguagem HTML.



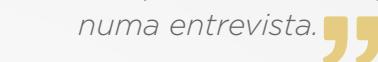
23:48



30:25



*“ Explicar o porquê que resolve é o que diferencia alguém numa entrevista.”*



## AULA 2 • PARTE 2

### CSS e acesso a pastas

Os sites em geral são baseados em links, um site linka para outro, até mesmo para baixar conteúdos do próprio site que está sendo acessado. O professor Mário Souto demonstra como isso se aplica na programação em CSS, de forma prática.

No dia a dia da programação é comum ter códigos que acessam pastas e mexem em arquivos, logo é importante saber os comandos como ls e cd, por exemplo, para buscas e acessar sistemas.



00:00

08:37



### Praticando

O professor Mário Souto apresenta um exemplo prático de como simular um navegador por meio de código. Destaca que quando queremos agrupar informações criamos um objeto. Demonstra a criação de objetos, ou coisas com múltiplos atributos, e arrays, que são basicamente listas.

18:37



### Praticando: continuação

Seguindo o exemplo, o professor Mário Souto demonstra como apresentar o código realizado de forma visual. Para isso o navegador possui uma estrutura chamada document, que lê a página, que apresenta as estruturas que podem ser usadas para trazer o código a forma visual. O professor aplica ainda os conhecimentos vistos anteriormente.

## EXERCÍCIO DE FIXAÇÃO

É uma das peculiaridades dos objetos do JavaScript:



## AULA 2 • PARTE 3

### PALAVRAS-CHAVE

**hacker rank:** Solução em avaliação técnica e entrevista remota para contratação de desenvolvedores.

Acesse em: <https://www.hackerrank.com/>

### PALAVRAS-CHAVE

**Lua:** Linguagem de programação interpretada, de script em alto nível, com tipagem dinâmica e multiparadigma, reflexiva e leve, projetada por Tecgraf da PUC-Rio em 1993 para expandir aplicações em geral, de forma extensível, para prototipagem e para ser embarcada em softwares complexos.

“ Todo código que estamos escrevendo, vamos sempre deixar mais descritivo. ”

01:53



### Estruturas e demanda

O professor Mário Souto destaca que o profissional da área estará sempre criando estruturas, que muitas vezes, a nível de código, serão diferentes devido ao tipo de demanda que se recebe. E apresenta um exercício prático de criação de lista (Queue), lembrando que um dado entrará após o outro (a ordem importa) e quem entra primeiro sai primeiro.

06:19



### EXERCÍCIO DE FIXAÇÃO

É uma particularidade do array:

09:47



10:37



### Array, fila e objeto

Várias linguagens tem alguma estrutura que pode ser interna ou alguma variação, para manter várias coisas guardadas no espaço de memória e criar filas.

15:02



A diferença entre array e a estrutura de dados fila, é que quando se tem uma estrutura criamos representações que facilitam a reprodução dessa. Ou seja, é como se fosse uma receita de criação daquela estrutura.

Na maioria das linguagens, para criar um objeto também existe uma receita para criação, pois um objeto é um conjunto de atributos e métodos. Essa receita de criação de objetos, ou estrutura padronizada, se chama classe.

## EXERCÍCIO DE FIXAÇÃO

Qual a estrutura de dados que representa uma receita de criação de objetos?

22:24



“Normalmente, qualquer problema que você vai resolver, alguém já resolveu antes e provavelmente tem uma solução ótima para resolver isso.”

## PALAVRAS-CHAVE

**Refactoring:** Livro de Martin Fowler, que explica a Refatoração. Aprenda a aperfeiçoar o design e a manutenção de códigos existentes, deixando-os mais fácil de entender.

22:47



### Design first e princípios

Pensar em como será o design do código em primeiro lugar, deixa mais legível como se irá trabalhar com ele, possibilitando ajustes posteriores para se adaptar à necessidade do momento.

29:08



Um dos princípios muito comuns de programação é o open-closed, aberto para extensão e fechado para modificação, isso evita que sejam feitas alterações no código.

## AULA 2 • PARTE 4

### Histórico do navegador

As setas de histórico do navegador guardam todos os sites navegados em determinada aba, possibilitando ir para um específico, retornar em sites já visitados e apagar os sites quando retornado e visitado um novo site. Executando o processo inverso ao FIFO (first in first out), conhecido como LIFO (last in first out).

Esse histórico é uma estrutura chamada de pilha (stack) que, como o nome já diz, vai empilhar itens.

03:10



### PALAVRA-CHAVE

**Stackoverflow:** Erro ocorre quando um programa tenta usar mais espaço do que o disponível na pilha de chamadas, chamado de “estouro de pilha”, normalmente resultando em uma falha do programa.

## Praticando

O professor Mário Souto demonstra de forma prática como a estrutura de pilha do histórico do navegador funciona. Traz que a pilha trabalha o conceito LIFO, mas um histórico completo de navegador é uma estrutura muito mais complexa que essa, com diversas outras funções. E destaca que olhar algo que acontece debaixo dos panos e tentar entender como funciona é muito importante no aprendizado de programação.

## PALAVRAS-CHAVE

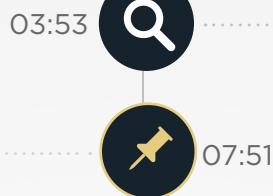
**MDN Web Docs:** O site MDN Web Docs fornece informações sobre tecnologias Open Web, incluindo HTML, CSS e APIs para sites e aplicativos da Web progressivos.

Acesse em: <https://developer.mozilla.org/pt-BR/>

## Map e set

O map é uma estrutura de dados pronta que é basicamente uma lista, mas para além disso, o map tem as funcionalidades necessárias para manipular um grupo de objetos onde cada elemento é um par que associa um objeto chave a um objeto valor. O professor Mário Souto demonstra de forma prática como essa estrutura funciona. Destaca que o map além de resolver prioridade, podemos acessar um item pela sua chave, otimizando o acesso à informação.

Outra estrutura que otimiza a experiência do usuário é o Set, que aceita valores únicos, excluindo automaticamente valores duplicados. O map e o set tem basicamente o mesmo princípio de sobrescrição, porém o map sobrescreve com base no último item, e o set evita múltiplas cópias do mesmo item.



## PALAVRA-CHAVE

**Loupe:** Site que disponibiliza é uma pequena visualização para ajudar desenvolvedores a entender como a pilha de chamadas/loop de eventos/fila de retorno de chamada do JavaScript interage entre si.

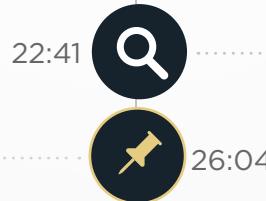
Acesse em: <http://latentflip.com/loupe/>



## Test driven development

Entender o que se está fazendo já é meio caminho andado para fazer o código funcionar. Nesse sentido, uma técnica muito utilizada na programação é o desenvolvimento guiado por testes, que se baseia em um ciclo curto de repetições de verificação e validação. O ciclo é definido por escrever o código, fazer ele funcionar, para então refaturar.

Esse processo de escrever e testar ajuda a ter consciência do que se está fazendo, escrever muito e não testar, torna difícil de encontrar o que está fazendo de errado.



## PALAVRA-CHAVE

**Quokka:** Site que executa um código JavaScript e TypeScript e exibe os resultados em linha no VS Code, WebStorm, Sublime Text e Atom.

Acesse em: <https://quokkajs.com/>

## Compartilhando

Uma forma de compartilhar o seu trabalho é por meio do Github. O professor Mário Souto demonstra como compartilhar seu código desenvolvido por meio da plataforma, destacando que além de contribuir para a comunidade da área, seu histórico fica visível para futuros empregadores. O compartilhamento do Github também é muito útil para trabalhar em grupo no mesmo código.



37:14

## EXERCÍCIO DE FIXAÇÃO

Quais as estruturas que tem o princípio de sobreescrita?

45:58



**“**A nossa área só cresce muito porque o conhecimento é aberto.**”**

## AULA 3 • PARTE 1

### Pensamento computacional

É importante pensar em exatamente como uma função é construída dentro, em um pensamento computacional existe um caminho que temos que seguir para conduzir uma solução. A partir da apresentação de um determinado problema, seguimos para a exploração desse e entendimento das características de funcionamento deste problema para que, com recursos computacionais e conhecimento adquirido, possamos efetivamente traçar caminhos e definir a melhor solução a ser aplicada.

Antes de começar a desenvolver a programação, é importante que se tenha em mente exatamente qual é o problema que quer solucionar, qual a forma de condução da solução e recursos.



01:14

02:03



**“**Para um problema, existem necessariamente várias funções que podem ser aplicadas, nem todas são necessariamente boas.**”**

04:49



## PALAVRA-CHAVE

**Node.js:** Software de código aberto, multiplataforma, baseado no interpretador V8 do Google e que permite a execução de códigos JavaScript fora de um navegador web.

## Introdução

O professor Edson Moreno apresenta um código simples de escrita na tela, para introduzir os softwares e a linguagem que serão utilizados no decorrer da aula. Retoma o conceito de JavaScript, que é uma linguagem de programação baseada em objetos, dinâmica, fracamente tipada e usualmente interpretada por navegadores. Com ela, pode-se criar scripts que manipulam HTML e CSS, mas também podem ser criados scripts para serem executados no server side (backend). A caracterização do arquivo é uma extensão .js.

O ponto de partida de qualquer linguagem de programação é o entendimento de como eu criei variáveis. O JavaScript permite três tipos de caracterização de variáveis: let, const e var. Outro ponto importante é entender os comandos básicos que podem ser utilizados pela linguagem, que vão participar basicamente de comandos de repetição (for, while e do while), de seleção (if, else, switch) e modulação.



05:55

## EXERCÍCIO DE FIXAÇÃO

Qual a extensão de um arquivo JavaScript?



20:45

## Modularização e exemplo

A modularização vem da criação de funções de recursos que vão estar disponíveis de forma concentrada dentro do código. Uma vez chamada essa função, existe um conjunto de comandos que estão associados a essa função especificamente, que serão executados e irão retornar ou não um valor.

O professor Edson Moreno apresenta um código simples para demonstrar a aplicação dos componentes da linguagem JavaScript vistos até o momento.



30:33

## Complexidade algorítmica

Quando criamos uma solução computacional, muitas vezes, é necessário parar e analisar se a solução é a melhor. Resolver um problema não significa que a solução é boa.

Para saber se um algoritmo é eficiente é necessário avaliar algumas características de desempenho como o espaço ocupado e o tempo gasto na execução.

A complexidade de um algoritmo é a medida de consumo de recursos com o tempo de processamento, memória que ocupada, largura de banda de comunicação, hardware necessário, etc.



31:02

 A formação que a gente busca do desenvolvedor full stack é mais do que pura e simplesmente a junção de comandos. 

Uma das formas comuns de saber se uma determinada solução é adequada ou não, é o cálculo de tempo. Mas quando nos preocupamos tão somente com o tempo de uma determinada computação, essa pode não ser a melhor forma de analisar. Pois o nosso sistema computacional não está dedicado exclusivamente para aquilo que queremos executar, existe um universo de outras coisas que estão rodando no computador, que influenciam no tempo de computação.

“ Muitas coisas adversas, alheias à nossa vontade podem acontecer dentro do momento de execução do nosso programa. ”

## EXERCÍCIO DE FIXAÇÃO

O que é uma solução primitiva?

36:42



37:05



## Avaliação de desempenho

Como visto, o cálculo de tempo pode não ser uma boa abordagem para avaliar o desempenho de um programa. Então, outra solução é mensurar qual o melhor caso, o caso médio ou pior caso de uso de uma solução. Para isso, ao invés de contato tempo contamos operações, e a partir dessas formular uma equação que define qual é o comportamento do algoritmo.

Esse cálculo consiste em contar quantas operações primitivas são executadas, lembrando que isso pode ser executado mais de uma vez. Além disso, para diferentes algoritmos teremos diferentes comportamentos, portanto precisamos entender a complexidade algorítmica para tentar extrair uma equação que possa medir o desempenho.

## PALAVRAS-CHAVE

**Grande-O:** A notação O-grande descreve o comportamento limitante de uma função quando o argumento tende a um valor específico ou para o infinito, normalmente, em termos de funções mais simples.

44:06



45:36



## PALAVRA-CHAVE

**Função quadrática:** Em álgebra, é uma função polinomial associada a um polinômio do segundo grau, então ela possui a mesma forma

## AULA 3 • PARTE 2

### EXERCÍCIO DE FIXAÇÃO

O array é um recurso indexado, isso significa que:

**Tudo é costume. É a forma como eu vou trabalhar de fato, como eu me sinto mais à vontade para fazer esse tipo de elaboração.**

#### Array: continuação

O conjunto de elementos que podem ser explorados na elaboração utilizando array são: length, push, pop, shift, unshift, splice, slice, at(pos) ou [pos], etc.

A partir do array e seus métodos é possível construir estruturas como pilhas e filas. Para a construção de uma pilha são necessários somente dois métodos, o push e o pop. No caso da fila utilizamos os métodos push e shift.

O professor Edson Moreno apresenta exemplos de construções de pilha e fila utilizando o array. E destaca que as estruturas básicas apresentadas são recursos importantes na criação das soluções em que não se tem métodos predefinidos.

00:52



#### Array

Uma das estruturas de dados padrão para linguagens é o array, que tem a função de armazenar dados de forma contínua na memória. Ele facilita percorrer os dados armazenados e evita a criação de múltiplas variáveis para o mesmo fim. O Array é um recurso indexado, ou seja, é possível acessar os elementos individuais dentro dele.

03:19



#### PALAVRA-CHAVE

**Funções Lambda:** São funções que o usuário não vai precisar escrever a função e depois utilizá-la dentro do código. Em JavaScript, a expressão Lambda permite definir uma interface funcional (novamente, um método abstrato) que o compilador identifica pela estrutura.

09:39



#### PALAVRA-CHAVE

**Sequência de Fibonacci:** É uma sequência de números inteiros, começando normalmente por 0 e 1, na qual cada termo subsequente corresponde à soma dos dois anteriores.

15:43



**Software, como o próprio nome diz, soft, nós fazemos a adequação que é necessária.**

## Maps

O maps é simples como um array, mas tem uma forma de acesso diferente. Essa estrutura também armazena os dados de forma contígua. Diferentemente do array, ela armazena um valor a partir de uma chave + valor, as chaves definidas dentro desse meio ambiente são únicas, mas o valor pode ser repetido.

A forma de interação com essa estrutura é o for, acessando cada posição para encontrar o valor. O professor Edson Moreno dá exemplos de aplicação desse método na estrutura.

25:48



### Set

Outra estrutura de dados simples presente nas linguagens é o set, ou conjuntos, que também armazena dados de forma contínua. Sua principal característica é que ele não permite que os dados armazenados nele sejam ou estejam duplicados.

32:59



O professor Edson Moreno demonstra de forma prática como a estrutura funciona, tentando forçar a repetição de valores. Assim como o array, o set também possui um conjunto de operações que são específicas da estrutura. Uma vez que alimentada, essa minha estrutura de dados, com algum dado específico, não é preciso sempre armazenar com o mesmo valor, mas ainda é possível armazenar com outras estruturas.

## AULA 3 • PARTE 3

### EXERCÍCIO DE FIXAÇÃO

Quais os dois atributos que compõem um nodo?

00:43



### Lista encadeada

Uma lista encadeada consiste num determinado número de nodos, cada um com uma referência para o próximo. São uma proposta de organização de uma estrutura mais complexa, trabalhando o conceito de referência, onde, além de uma estrutura de armazenamento vai existir uma referência de onde está o próximo elemento armazenado dentro dessa estrutura de dados.

O conceito de lista encadeada trabalha com nodos, que armazena a informação e tem a informação de onde está o próximo nodo. Ele possui dois atributos: element (dado armazenado) e next (referência ao objeto de mesma classe).

## Inserção de elementos

Para a inserção de elementos em listas encadeadas é necessário:

Alocação de um novo nodo;

Inserção das informações no nodo alocado;

Inserção de nodo na lista com consequentemente encadeamento nos nodos já existentes.

O professor Edson Moreno apresenta exemplos de inserção, substituição e remoção de elementos na estrutura, demonstrando o ganho de custo na manipulação dos elementos.

## Exemplos

O professor Edson Moreno demonstra exemplos da criação de um nodo e a implementação da estrutura de lista encadeada. A lista encadeada tem a grande vantagem, com relação a memória contíguas, de que toda vez que for necessária uma manipulação, adicionar um elemento no início ou adicionar um elemento no meio, por exemplo, não se tem a complexidade envolvida em memórias contigo de ajuste de memória, simplesmente se descarta o elemento desnecessário.

O professor traz ainda exemplos de estruturas encadeadas utilizando o conceito aprendido. Destaca que o uso somente da referência header é ineficiente para a inserção de elementos no final da lista.

## PALAVRA-CHAVE

## Árvores

Uma árvore é uma estrutura não linear que permite a implementação de vários algoritmos mais rápidos do que estruturas de dados lineares, como as listas, por exemplo, e fornece uma forma natural de organizar os dados.

Essa é uma estrutura que privilegia uma construção hierárquica, e precisamos ter o entendimento de como funciona para tirar o melhor proveito desse tipo de estrutura. Outra propriedade de uma árvore qualquer é que cada nodo tem um único nodo pai, ou seja, o número de elementos que se referencia para baixo na árvore é uma decisão estrutural.

Por definição uma árvore pode ser:

Uma árvore pode ser vazia, quando não tem nenhum nodo;

Isso permite a definição de uma árvore recursivamente;

Ou tenho uma árvore sem nodos, ou tenho uma relação tal que um nodo tem os seus filhos e cada filho tem o seu pai nessa construção.



## Recursão

Para trabalhar com a árvore existe uma forma de construção de algoritmo que é mais adequado para esse tipo de estrutura. A recursão é um método que consiste em solucionar partes menores do mesmo problema e finaliza quando o problema original foi resolvido.

O professor Edson Moreno apresenta exemplos da utilização desse método. Esse método se aplica a árvores, visto que a árvore é uma estrutura em que fazer a navegação entre os elementos enquanto tiver um elemento disponível, é uma oportunidade de navegação na qual a recursão se encaixa.



**Fatorial:** Notação introduzida por Christian Kramp, em 1808, o fatorial de um número natural  $n$ , representado por  $n!$ , é o produto de todos os inteiros positivos menores ou iguais a  $n$ .

## AULA 3 • PARTE 4

00:03



### Árvores: continuação

A estrutura da árvore possui algumas características como:

Grau: define quantos elementos apontam a partir de um nodo;

Nível do nodo: em qual camada da minha árvore o nodo se encontra;

Altura: quantos níveis compõem a estrutura da árvore.

Alguns métodos são esperados sempre quando a gente vai trabalhar com uma árvore, para questões de navegação, possibilitando realizar consultas de elementos e modificações.

As estruturas de árvores podem ser construídas de diferentes formas, mas o mais adequado e mais comum, são relações onde temos coisas existindo nos diferentes espaços da memória.

Uma árvore binária é uma árvore organizada com as seguintes propriedades: todos os nodos têm no máximo dois filhos, cada nodo filho é rotulado como ser um filho da esquerda ou um filho da direita, e o filho da esquerda precede o filho da direita na ordenação dos filhos de um nodo.

## EXERCÍCIO DE FIXAÇÃO

É armazenado pela estrutura de árvore:



## Árvores: continuação

Mais do que garantir a estrutura da árvore, é importante ser possível fazer a navegação sobre esse tipo de recurso.

Quando trabalhamos com caminhamento em profundidade existem três formas de fazer esse processo:

Pré-fixado: nodo é visitado antes de seus descendentes, segue a ordem visita a raiz, percorre a subárvore da esquerda seguindo para a da direita;

Pós-fixado: nodo é visitado depois de seus descendentes, segue a ordem percorre a subárvore da esquerda seguindo para a da direita e visitando a raiz;

Central: segue a ordem percorre a subárvore da esquerda, visita a raiz e então percorre a da direita.

Quando trabalhamos com caminhamento em largura, o método visita os nodos na ordem dos níveis da árvore, da esquerda para a direita.

**“**A ausência de uma organização melhor tem um impacto grande na nossa solução.**”**

## Algoritmos de busca

Uma vez que a solução está ordenada, podemos buscar a informação. Existem basicamente duas formas de fazer isso:

Busca linear: onde se busca um elemento após o outro, investigando cada posição da estrutura de dados e retornando o item solicitado;

Busca binária: requer que a estrutura esteja ordenada e sorteia uma posição, se o valor for maior do que o procurado então sorteia a esquerda. Se o valor for menor que o procurado sorteia a direita.



10:36



16:44

## PALAVRA-CHAVE

**Adrian Mejia:** Engenheiro de software e eletrônica, desenvolvedor web full-stack, autor, atualmente engenheiro de software no Google em Boston, EUA.



17:24

## Mundo dos algoritmos

São alguns exemplos de algoritmos disponíveis:

Divisão e conquista: é uma solução do qual quebra-se a solução em tamanhos menores para que o problema se torne muito fácil de ser resolvido;

Programação dinâmica: quebra do problema em partes menores, mas independentes;

Guloso: seu grande objetivo é entrar num ponto tal de refinamento encontrando o ponto ótimo;

Algoritmos de Backtracking: vai buscar uma solução com refinamento, mas no momento que percebe que não há mais evolução, faz um retorno para um ponto onde entende que teve um bom ganho, e começa a buscar uma outra solução daquele ponto.



23:05



31:19

Quando trabalhamos com essas estruturas de dados que buscam a melhora de desempenho sobre essas funções, temos que ter organização. A presença de dados desordenados impactará no desempenho da solução.

A entrega de uma solução desorganizada pode acontecer, e para isso existem alguns algoritmos de ordenamento: bubble sort, selection sort, insertion sort, merge sort e quicksort.

# Resumo da disciplina

Veja, nesta página, um resumo dos principais conceitos vistos ao longo da disciplina.

## AULA 1

A base vai muito de se questionar como as coisas funcionam.



Quanto mais automático, mais você paga para alguém fazer, quanto menos automático mais o time da sua empresa vai ter que fazer.

Qualquer coisa que você faz no computador é código rodando.



## AULA 2

Normalmente, qualquer problema que você vai resolver, alguém já resolveu antes e provavelmente tem uma solução ótima para resolver isso.



Um mesmo problema pode ser resolvido de formas diferentes, e a forma que você abstrai a informação vai depender do problema que te deram.



A nossa área só cresce muito porque o conhecimento é aberto.

## AULA 3

Para um problema, existem necessariamente várias funções que podem ser aplicadas. Nem todas são necessariamente boas.



Software, como o próprio nome diz, soft, nós fazemos a adequação que é necessária.

A ausência de uma organização melhor tem um impacto grande na nossa solução.



# Avaliação

Veja as instruções para realizar a avaliação da disciplina.

Já está disponível o teste online da disciplina. O prazo para realização é de **dois meses a partir da data de lançamento das aulas**.

Lembre-se que cada disciplina possui uma avaliação online.  
A nota mínima para aprovação é 6.

Fique tranquilo! Caso você perca o prazo do teste online, ficará aberto o teste de recuperação, que pode ser realizado até o final do seu curso. A única diferença é que a nota máxima atribuída na recuperação é 8.

