TI MAPEADA · 300 MAPAS MENTAIS DE TI

Módulo 02

Desenvolvimento e Programação

30 mapas · mapas 032 a 061



Neste módulo:

032 Paradigmas de Programação
033 POO: os 4 Pilares
034 Classe, Objeto e Interface
035 Sobrecarga e Sobrescrita
036 Princípios SOLID
037 Padrões GoF: Criacionais
038 Padrões GoF: Estruturais
039 Padrões GoF: Comportamentais
040 Estruturas de Dados Lineares
041 Árvores
042 Grafos
043 Algoritmos de Ordenação
044 Complexidade e Busca
045 Recursão
046 Java: Fundamentos
047 Java: Coleções
048 Python: Fundamentos
049 JavaScript: Fundamentos
050 HTML5
051 CSS3
052 HTTP: Métodos e Status
053 APIs REST
054 SOAP e Web Services
055 JSON e XML
056 Microsserviços
057 MVC e Arquitetura em Camadas
058 Frameworks Web
059 Clean Code
060 Acessibilidade Web
061 Desenvolvimento Mobile

Paradigmas de Programação

O QUE É

Estilos de organizar e escrever programas.

1 Imperativo

- sequência de comandos que mudam o estado
- C
- Pascal

2 Orientado a objetos

- classes e objetos
- Java
- C#

3 Funcional

- funções puras
- imutabilidade
- Haskell
- Lisp

4 Lógico

- fatos e regras
- Prolog

5 Multiparadigma

- Python
- JavaScript
- Scala

⚠ CAI EM PROVA

Na programação funcional pura não há efeitos colaterais.

Material de estudo e revisão · uso pessoal

POO: os 4 Pilares

O QUE É

Princípios que sustentam a programação orientada a objetos.

1 Abstração

- foca no essencial do objeto

2 Encapsulamento

- esconde detalhes
- acesso por métodos

3 Herança

- subclasse reaproveita a superclasse

4 Polimorfismo

- mesma mensagem, comportamentos diferentes

5 Apoio

- coesão alta
- acoplamento baixo

⚠ CAI EM PROVA

Encapsulamento protege o estado interno usando modificadores de acesso.

Material de estudo e revisão · uso pessoal

Classe, Objeto e Interface

O QUE É

Elementos básicos para modelar software orientado a objetos.

1 Classe

- molde com atributos e métodos

2 Objeto

- instância concreta de uma classe

3 Interface

- contrato de métodos sem estado

4 Classe abstrata

- não pode ser instanciada
- pode ter código

5 Construtor

- inicializa o objeto

⚠ CAI EM PROVA

Em Java uma classe herda de uma só classe, mas implementa várias interfaces.

Material de estudo e revisão · uso pessoal

Sobrecarga e Sobrescrita

O QUE É

Duas formas de polimorfismo em linguagens orientadas a objetos.

1 Sobrecarga

- mesmo nome
- parâmetros diferentes
- mesma classe

2 Tipo

- polimorfismo estático, em tempo de compilação

3 Sobrescrita

- subclasse redefine método da superclasse

4 Tipo

- polimorfismo dinâmico, em tempo de execução

5 Java

- anotação @Override na sobrescrita

⚠ CAI EM PROVA

Mudar só o tipo de retorno não caracteriza sobrecarga.

Princípios SOLID

O QUE É

Cinco princípios de design orientado a objetos de Robert C. Martin.

1 S

- responsabilidade única
- um motivo para mudar

2 O

- aberto para extensão
- fechado para modificação

3 L

- substituição de Liskov
- subclasse substitui a base

4 I

- segregação de interfaces
- interfaces pequenas

5 D

- inversão de dependência
- depender de abstrações

⚠ CAI EM PROVA

O princípio Aberto/Fechado prega estender sem alterar o código existente.

Padrões GoF: Criacionais

O QUE É

Padrões que tratam da criação de objetos.

1 Singleton

- uma única instância global

2 Factory Method

- subclasses decidem o que criar

3 Abstract Factory

- famílias de objetos relacionados

4 Builder

- constrói objetos complexos passo a passo

5 Prototype

- cria por cópia de um protótipo

⚠ CAI EM PROVA

O GoF tem 23 padrões: 5 criacionais, 7 estruturais e 11 comportamentais.

Padrões GoF: Estruturais

O QUE É

Padrões que tratam da composição de classes e objetos.

1 Adapter

- converte uma interface em outra

2 Facade

- interface simples para um subsistema

3 Decorator

- acrescenta responsabilidades dinamicamente

4 Composite

- trata parte e todo de forma igual

5 Outros

- Proxy
- Bridge
- Flyweight

⚠ CAI EM PROVA

Proxy controla o acesso; Decorator acrescenta comportamento.

Padrões GoF: Comportamentais

O QUE É

Padrões que tratam da comunicação e das responsabilidades entre objetos.

1 Observer

- notifica dependentes quando o estado muda

2 Strategy

- troca algoritmos em tempo de execução

3 Command

- encapsula uma requisição como objeto

4 Template Method

- esqueleto do algoritmo na superclasse

5 Outros

- State
- Iterator
- Mediator
- Chain of Responsibility

⚠ CAI EM PROVA

Observer é a base do modelo publicar e assinar.

Material de estudo e revisão · uso pessoal

Estruturas de Dados Lineares

O QUE É

Organizam elementos em sequência.

1 Vetor

- acesso direto por índice
- tamanho fixo

2 Lista encadeada

- nós com ponteiros
- inserção fácil

3 Pilha

- LIFO
- push e pop

4 Fila

- FIFO
- enqueue e dequeue

5 Deque

- insere e remove nas duas pontas

⚠ CAI EM PROVA

Pilha é LIFO (último a entrar, primeiro a sair); fila é FIFO.

Material de estudo e revisão · uso pessoal

Árvores

O QUE É

Estrutura hierárquica com nó raiz, nós internos e folhas.

1 Binária

- no máximo 2 filhos por nó

2 Busca binária

- esquerda menor
- direita maior

3 AVL

- balanceada pela altura
- fator entre -1 e 1

4 B e B+

- muitas chaves por nó
- usadas em bancos de dados

5 Percursos

- pré-ordem
- em ordem
- pós-ordem

⚠ CAI EM PROVA

O percurso em ordem de uma árvore de busca binária sai ordenado.

Grafos

O QUE É

Conjunto de vértices ligados por arestas.

1 Tipos

- direcionado
- não direcionado
- ponderado

2 Representação

- matriz de adjacência
- lista de adjacência

3 Busca

- BFS em largura
- DFS em profundidade

4 Menor caminho

- Dijkstra
- Bellman-Ford

5 Árvore geradora mínima

- Prim
- Kruskal

⚠ CAI EM PROVA

Dijkstra não funciona com arestas de peso negativo.

Algoritmos de Ordenação

O QUE É

Métodos para colocar elementos em ordem.

1 Simples O(n²)

- bolha
- seleção
- inserção

2 Eficientes O(n log n)

- merge sort
- heap sort

3 Quick sort

- médio O(n log n)
- pior O(n²)

4 Estável

- mantém a ordem dos iguais
- merge
- inserção

5 Inserção

- ótimo para vetores quase ordenados

⚠ CAI EM PROVA

Quick sort tem pior caso O(n²), apesar de ser rápido na média.

Material de estudo e revisão · uso pessoal

Complexidade e Busca

O QUE É

Mede como tempo e memória crescem com o tamanho da entrada.

1 Notação Big O

- limite superior do crescimento

2 O(1)

- constante
- acesso por índice

3 O(log n)

- busca binária em vetor ordenado

4 O(n)

- busca sequencial

5 Hash

- busca média O(1)

⚠ CAI EM PROVA

Busca binária exige o vetor ordenado.

Material de estudo e revisão · uso pessoal

Recursão

O QUE É

Função que chama a si mesma para resolver subproblemas menores.

1 Caso base

- condição de parada obrigatória

2 Caso recursivo

- reduz o problema

3 Exemplos

- fatorial
- Fibonacci
- torres de Hanói

4 Risco

- estouro de pilha sem caso base

5 Alternativa

- versão iterativa com laço

⚠ CAI EM PROVA

Toda recursão precisa de caso base para terminar.

Java: Fundamentos

O QUE É

Linguagem orientada a objetos compilada para bytecode e executada na JVM.

1 Plataforma

- JDK desenvolve
- JRE executa
- JVM roda o bytecode

2 Tipos primitivos

- byte
- short
- int
- long
- float
- double
- char
- boolean

3 Modificadores

- public
- protected
- default
- private

4 Exceções

- checked exigem tratamento
- unchecked não

5 Memória

- garbage collector libera objetos sem referência

⚠ CAI EM PROVA

String em Java é imutável; para concatenar muito, use StringBuilder.

Java: Coleções

O QUE É

Framework de estruturas de dados prontas do pacote java.util.

1 List

- ordenada
- aceita repetidos
- ArrayList
- LinkedList

2 Set

- sem repetidos
- HashSet
- TreeSet

3 Map

- chave e valor
- HashMap
- TreeMap

4 Queue

- fila
- PriorityQueue

5 Tree

- mantém elementos ordenados

⚠ CAI EM PROVA

Map não estende a interface Collection.

Material de estudo e revisão · uso pessoal

Python: Fundamentos

O QUE É

Linguagem interpretada, de tipagem dinâmica e forte, com indentação obrigatória.

1 Tipos

- int
- float
- str
- bool
- None

2 Coleções

- list mutável
- tuple imutável
- dict
- set

3 Estruturas

- if
- for
- while
- def
- class

4 Recursos

- list comprehension
- slicing
- lambda

5 Ecossistema

- pip
- bibliotecas como Pandas e NumPy

⚠ CAI EM PROVA

Tupla é imutável; lista é mutável.

Material de estudo e revisão · uso pessoal

JavaScript: Fundamentos

O QUE É

Linguagem da web, interpretada, de tipagem dinâmica e fraca.

1 Variáveis

- var de função
- let e const de bloco

2 Comparação

- == converte tipos
- === compara tipo e valor

3 Funções

- arrow functions
- callbacks
- closures

4 Assíncrono

- Promise
- async e await
- event loop

5 Onde roda

- navegador
- Node.js no servidor

⚠ CAI EM PROVA

const impede reatribuição, mas não torna objetos imutáveis.

HTML5

O QUE É

Linguagem de marcação que estrutura o conteúdo das páginas web.

1 Semântica

- header
- nav
- main
- article
- section
- footer

2 Mídia

- video
- audio
- canvas
- svg

3 Formulários

- novos tipos email
- date
- number
- required

4 APIs

- localStorage
- geolocalização
- Web Workers

5 Doctype

- declaração simples <!DOCTYPE html>

⚠ CAI EM PROVA

Tags semânticas melhoram acessibilidade e SEO.

CSS3

O QUE É

Linguagem de estilos que define a aparência das páginas.

1 Seletores

- elemento
- .classe
- #id
- atributo

2 Especificidade

- inline > id > classe > elemento

3 Box model

- conteúdo
- padding
- border
- margin

4 Layout

- Flexbox em uma dimensão
- Grid em duas

5 Responsivo

- media queries
- unidades relativas

⚠ CAI EM PROVA

!important sobrepõe a especificidade normal.

HTTP: Métodos e Status

O QUE É

Protocolo de aplicação, sem estado, de requisição e resposta.

1 GET

- obtém recurso
- idempotente

2 POST

- cria ou processa
- não idempotente

3 PUT e DELETE

- substituem e removem
- idempotentes

4 2xx e 3xx

- sucesso
- redirecionamento

5 4xx e 5xx

- erro do cliente
- erro do servidor

⚠ CAI EM PROVA

404 é recurso não encontrado; 500 é erro interno do servidor.

APIs REST

O QUE É

Estilo arquitetural para serviços web baseado em recursos e HTTP.

1 Restrições

- cliente-servidor
- stateless
- cache
- interface uniforme

2 Mais

- sistema em camadas
- código sob demanda (opcional)

3 Recurso

- identificado por URI

4 Formatos

- JSON
- XML

5 Maturidade de Richardson

- níveis 0 a 3
- HATEOAS

⚠ CAI EM PROVA

REST é estilo arquitetural, não protocolo.

Material de estudo e revisão · uso pessoal

SOAP e Web Services

O QUE É

Protocolo de troca de mensagens XML entre sistemas.

1 Envelope SOAP

- header e body

2 WSDL

- descreve o serviço

3 UDDI

- diretório de serviços

4 Transporte

- geralmente HTTP
- pode usar SMTP

5 Padrões WS-*

- WS-Security
- WS-ReliableMessaging

⚠ CAI EM PROVA

SOAP usa sempre XML; REST aceita vários formatos.

Material de estudo e revisão · uso pessoal

JSON e XML

O QUE É

Formatos de texto para troca de dados entre sistemas.

1 JSON

- chave e valor
- leve
- nativo do JavaScript

2 Tipos JSON

- string
- número
- booleano
- null
- objeto
- array

3 XML

- tags
- atributos
- namespaces

4 Validação

- JSON Schema
- XSD e DTD

5 Consulta XML

- XPath
- XSLT transforma

⚠ CAI EM PROVA

JSON não aceita comentários nem vírgula sobrando no final.

Microsserviços

O QUE É

Arquitetura com serviços pequenos, independentes e implantáveis separadamente.

1 Características

- um serviço por capacidade de negócio
- banco próprio

2 Comunicação

- REST
- gRPC
- mensageria

3 Padrões

- API Gateway
- service discovery
- circuit breaker

4 Vantagens

- escala independente
- times autônomos

5 Desafios

- complexidade distribuída
- consistência eventual

⚠ CAI EM PROVA

Monólito modular é alternativa válida e mais simples para sistemas pequenos.

Material de estudo e revisão · uso pessoal

MVC e Arquitetura em Camadas

O QUE É

Separação de responsabilidades entre dados, apresentação e controle.

1 Model

- dados e regras de negócio

2 View

- apresentação ao usuário

3 Controller

- recebe entradas e coordena

4 Variações

- MVP
- MVVM

5 Camadas

- apresentação
- negócio
- persistência

⚠ CAI EM PROVA

No MVC a View não deve conter regra de negócio.

Frameworks Web

O QUE É

Bibliotecas e estruturas que aceleram o desenvolvimento de aplicações.

1 Back-end Java

- Spring Boot
- Jakarta EE

2 Back-end outros

- Django e Flask (Python)
- Express (Node.js)

3 Front-end

- React
- Angular
- Vue

4 ORM

- Hibernate
- JPA
- SQLAlchemy

5 Injeção de dependência

- núcleo do Spring

⚠ CAI EM PROVA

React é biblioteca de interface; Angular é framework completo.

Clean Code

O QUE É

Práticas para escrever código legível, simples e fácil de manter.

1 Nomes

- significativos e pronunciáveis

2 Funções

- pequenas
- fazem uma coisa só

3 Comentários

- só quando o código não explica

4 DRY

- não repita a si mesmo

5 KISS e YAGNI

- simples
- não faça o que não precisa

⚠ CAI EM PROVA

Code smell indica possível problema, não necessariamente um bug.

Acessibilidade Web

O QUE É

Garante que pessoas com deficiência consigam usar sites e sistemas.

1 WCAG princípios

- perceptível
- operável
- compreensível
- robusto

2 Níveis WCAG

- A
- AA
- AAA

3 eMAG

- modelo do governo federal brasileiro

4 Práticas

- texto alternativo
- contraste
- navegação por teclado

5 WAI-ARIA

- atributos de papel e estado

⚠ CAI EM PROVA

Os 4 princípios da WCAG formam a sigla POUR.

Desenvolvimento Mobile

O QUE É

Abordagens para criar aplicativos para celulares e tablets.

1 Nativo

- Kotlin no Android
- Swift no iOS

2 Híbrido e multiplataforma

- Flutter
- React Native

3 PWA

- web que funciona como app
- service worker

4 Vantagem nativa

- desempenho e acesso total ao hardware

5 Multiplataforma

- um código para vários sistemas

⚠ CAI EM PROVA

PWA usa service worker para funcionar offline.

Material de estudo e revisão · uso pessoal