TJ-SP

TRIBUNAL DE JUSTIÇA DE SÃO PAULO

Analista de Sistemas Judiciário

EDITAL Nº 01/2025

CÓD: OP-073FV-25
7908403570256

Língua Portuguesa

Conhecimentos Específicos
Analista de Sistemas Judiciário

Raciocínio Lógico Matemático

Legislação

LEITURA E INTERPRETAÇÃO DE TEXTOS DE DIVERSOS GÊNEROS DISCURSIVOS; RECONHECIMENTO DE INFORMAÇÕES IMPLÍCITAS E INFERÊNCIAS TEXTUAIS

A leitura e interpretação de textos são habilidades essenciais no âmbito dos concursos públicos, pois exigem do candidato a capacidade de compreender não apenas o sentido literal, mas também as nuances e intenções do autor. Os textos podem ser divididos em duas categorias principais: literários e não literários. A interpretação de ambos exige um olhar atento à estrutura, ao ponto de vista do autor, aos elementos de coesão e à argumentação. Neste contexto, é crucial dominar técnicas de leitura que permitam identificar a ideia central do texto, inferir informações implícitas e analisar a organização textual de forma crítica e objetiva.

— Compreensão Geral do Texto

A compreensão geral do texto consiste em identificar e captar a mensagem central, o tema ou o propósito de um texto, sejam eles explícitos ou implícitos. Esta habilidade é crucial tanto em textos literários quanto em textos não literários, pois fornece ao leitor uma visão global da obra, servindo de base para uma interpretação mais profunda. A compreensão geral vai além da simples decodificação das palavras; envolve a percepção das intenções do autor, o entendimento das ideias principais e a identificação dos elementos que estruturam o texto.

– Textos Literários

Nos textos literários, a compreensão geral está ligada à interpretação dos aspectos estéticos e subjetivos. É preciso considerar o gênero (poesia, conto, crônica, romance), o contexto em que a obra foi escrita e os recursos estilísticos utilizados pelo autor. A mensagem ou tema de um texto literário muitas vezes não é transmitido de maneira direta. Em vez disso, o autor pode utilizar figuras de linguagem (metáforas, comparações, simbolismos), criando camadas de significação que exigem uma leitura mais interpretativa.

Por exemplo, em um poema de Manuel Bandeira, como "O Bicho", ao descrever um homem que revirava o lixo em busca de comida, a compreensão geral vai além da cena literal. O poema denuncia a miséria e a degradação humana, mas faz isso por meio de uma imagem que exige do leitor sensibilidade para captar essa crítica social indireta.

Outro exemplo: em contos como "A Hora e a Vez de Augusto Matraga", de Guimarães Rosa, a narrativa foca na jornada de transformação espiritual de um homem. Embora o texto tenha uma história clara, sua compreensão geral envolve perceber os elementos de religiosidade e redenção que permeiam a narrativa, além de entender como o autor utiliza a linguagem regionalista para dar profundidade ao enredo.

– Textos Não Literários

Em textos não literários, como artigos de opinião, reportagens, textos científicos ou jurídicos, a compreensão geral tende a ser mais direta, uma vez que esses textos visam transmitir informações objetivas, ideias argumentativas ou instruções. Neste caso, o leitor precisa identificar claramente o tema principal ou a tese defendida pelo autor e compreender o desenvolvimento lógico do conteúdo.

Por exemplo, em um artigo de opinião sobre os efeitos da tecnologia na educação, o autor pode defender que a tecnologia é uma ferramenta essencial para o aprendizado no século XXI. A compreensão geral envolve identificar esse posicionamento e as razões que o autor oferece para sustentá-lo, como o acesso facilitado ao conhecimento, a personalização do ensino e a inovação nas práticas pedagógicas.

Outro exemplo: em uma reportagem sobre desmatamento na Amazônia, o texto pode apresentar dados e argumentos para expor a gravidade do problema ambiental. O leitor deve captar a ideia central, que pode ser a urgência de políticas de preservação e as consequências do desmatamento para o clima global e a biodiversidade.

– Estratégias de Compreensão

Para garantir uma boa compreensão geral do texto, é importante seguir algumas estratégias:

- Leitura Atenta: Ler o texto integralmente, sem pressa, buscando entender o sentido de cada parte e sua relação com o todo.

- Identificação de Palavras-Chave: Buscar termos e expressões que se repetem ou que indicam o foco principal do texto.

- Análise do Título e Subtítulos: Estes elementos frequentemente apontam para o tema ou ideia principal do texto, especialmente em textos não literários.

- Contexto de Produção: Em textos literários, o contexto histórico, cultural e social do autor pode fornecer pistas importantes para a interpretação do tema. Nos textos não literários, o contexto pode esclarecer o objetivo do autor ao produzir aquele texto, seja para informar, convencer ou instruir.

- Perguntas Norteadoras: Ao ler, o leitor pode se perguntar: Qual é o tema central deste texto? Qual é a intenção do autor ao escrever este texto? Há uma mensagem explícita ou implícita?

Exemplos Práticos

- Texto Literário: Um poema como "Canção do Exílio" de Gonçalves Dias pode, à primeira vista, parecer apenas uma descrição saudosista da pátria. No entanto, a compreensão geral deste texto envolve entender que ele foi escrito no contexto de um poeta exilado, expressando tanto amor pela pátria quanto um sentimento de perda e distanciamento.

- Texto Não Literário: Em um artigo sobre as mudanças climáticas, a tese principal pode ser que a ação humana é a principal responsável pelo aquecimento global. A compreensão geral exigiria que o leitor identificasse essa tese e as evidências apresentadas, como dados científicos ou opiniões de especialistas, para apoiar essa afirmação.

– Importância da Compreensão Geral

Ter uma boa compreensão geral do texto é o primeiro passo para uma interpretação eficiente e uma análise crítica. Nos concursos públicos, essa habilidade é frequentemente testada em questões de múltipla escolha e em questões dissertativas, nas quais o candidato precisa demonstrar sua capacidade de resumir o conteúdo e de captar as ideias centrais do texto.

Além disso, uma leitura superficial pode levar a erros de interpretação, prejudicando a resolução correta das questões. Por isso, é importante que o candidato esteja sempre atento ao que o texto realmente quer transmitir, e não apenas ao que é dito de forma explícita. Em resumo, a compreensão geral do texto é a base para todas as outras etapas de interpretação textual, como a identificação de argumentos, a análise da coesão e a capacidade de fazer inferências.

— Ponto de Vista ou Ideia Central Defendida pelo Autor

O ponto de vista ou a ideia central defendida pelo autor são elementos fundamentais para a compreensão do texto, especialmente em textos argumentativos, expositivos e literários. Identificar o ponto de vista do autor significa reconhecer a posição ou perspectiva adotada em relação ao tema tratado, enquanto a ideia central refere-se à mensagem principal que o autor deseja transmitir ao leitor.

Esses elementos revelam as intenções comunicativas do texto e ajudam a esclarecer as razões pelas quais o autor constrói sua argumentação, narrativa ou descrição de determinada maneira. Assim, compreender o ponto de vista ou a ideia central é essencial para interpretar adequadamente o texto e responder a questões que exigem essa habilidade.

– Textos Literários

Nos textos literários, o ponto de vista do autor pode ser transmitido de forma indireta, por meio de narradores, personagens ou símbolos. Muitas vezes, os autores não expõem claramente suas opiniões, deixando a interpretação para o leitor. O ponto de vista pode variar entre diferentes narradores e personagens, enriquecendo a pluralidade de interpretações possíveis.

Um exemplo clássico é o narrador de "Dom Casmurro", de Machado de Assis. Embora Bentinho (o narrador-personagem) conte a história sob sua perspectiva, o leitor percebe que o ponto de vista dele é enviesado, e isso cria ambiguidade sobre a questão central do livro: a possível traição de Capitu. Nesse caso, a ideia central pode estar relacionada à incerteza e à subjetividade das percepções humanas.

Outro exemplo: em "Vidas Secas", de Graciliano Ramos, o ponto de vista é o de uma narrativa em terceira pessoa que se foca nos personagens humildes e no sofrimento causado pela seca no sertão nordestino. A ideia central do texto é a denúncia das condições de vida precárias dessas pessoas, algo que o autor faz por meio de uma linguagem econômica e direta, alinhada à dureza da realidade descrita.

Nos poemas, o ponto de vista também pode ser identificado pelo eu lírico, que expressa sentimentos, reflexões e visões de mundo. Por exemplo, em "O Navio Negreiro", de Castro Alves, o eu lírico adota um tom de indignação e denúncia ao descrever as atrocidades da escravidão, reforçando uma ideia central de crítica social.

– Textos Não Literários

Em textos não literários, o ponto de vista é geralmente mais explícito, especialmente em textos argumentativos, como artigos de opinião, editoriais e ensaios. O autor tem o objetivo de convencer o leitor de uma determinada posição sobre um tema. Nesse tipo de texto, a tese (ideia central) é apresentada de forma clara logo no início, sendo defendida ao longo do texto com argumentos e evidências.

Por exemplo, em um artigo de opinião sobre a reforma tributária, o autor pode adotar um ponto de vista favorável à reforma, argumentando que ela trará justiça social e reduzirá as desigualdades econômicas. A ideia central, neste caso, é a defesa da reforma como uma medida necessária para melhorar a distribuição de renda no país. O autor apresentará argumentos que sustentem essa tese, como dados econômicos, exemplos de outros países e opiniões de especialistas.

Nos textos científicos e expositivos, a ideia central também está relacionada ao objetivo de informar ou esclarecer o leitor sobre um tema específico. A neutralidade é mais comum nesses casos, mas ainda assim há um ponto de vista que orienta a escolha das informações e a forma como elas são apresentadas. Por exemplo, em um relatório sobre os efeitos do desmatamento, o autor pode não expressar diretamente uma opinião, mas ao apresentar evidências sobre o impacto ambiental, está implicitamente sugerindo a importância de políticas de preservação.

– Como Identificar o Ponto de Vista e a Ideia Central

Para identificar o ponto de vista ou a ideia central de um texto, é importante atentar-se a certos aspectos:

1. Título e Introdução: Muitas vezes, o ponto de vista do autor ou a ideia central já são sugeridos pelo título do texto ou pelos primeiros parágrafos. Em artigos e ensaios, o autor frequentemente apresenta sua tese logo no início, o que facilita a identificação.

2. Linguagem e Tom: A escolha das palavras e o tom (objetivo, crítico, irônico, emocional) revelam muito sobre o ponto de vista do autor. Uma linguagem carregada de emoção ou uma sequência de dados e argumentos lógicos indicam como o autor quer que o leitor interprete o tema.

3. Seleção de Argumentos: Nos textos argumentativos, os exemplos, dados e fatos apresentados pelo autor refletem o ponto de vista defendido. Textos favoráveis a uma determinada posição tenderão a destacar aspectos que reforcem essa perspectiva, enquanto minimizam ou ignoram os pontos contrários.

4. Conectivos e Estrutura Argumentativa: Conectivos como "portanto", "por isso", "assim", "logo" e "no entanto" são usados para introduzir conclusões ou para contrastar argumentos, ajudando a deixar claro o ponto de vista do autor. A organização do texto em blocos de ideias também pode indicar a progressão da defesa da tese.

5. Conclusão: Em muitos textos, a conclusão serve para reafirmar o ponto de vista ou ideia central. Neste momento, o autor resume os principais argumentos e reforça a posição defendida, ajudando o leitor a compreender a ideia principal.

Exemplos Práticos

- Texto Literário: No conto "A Cartomante", de Machado de Assis, o narrador adota uma postura irônica, refletindo o ceticismo em relação à superstição. A ideia central do texto gira em torno da crítica ao comportamento humano que, por vezes, busca respostas mágicas para seus problemas, ignorando a racionalidade.

- Texto Não Literário: Em um artigo sobre os benefícios da alimentação saudável, o autor pode adotar o ponto de vista de que uma dieta equilibrada é fundamental para a prevenção de doenças e para a qualidade de vida. A ideia central, portanto, é que os hábitos alimentares influenciam diretamente a saúde, e isso será sustentado por argumentos baseados em pesquisas científicas e recomendações de especialistas.

– Diferença entre Ponto de Vista e Ideia Central

Embora relacionados, ponto de vista e ideia central não são sinônimos. O ponto de vista refere-se à posição ou perspectiva do autor em relação ao tema, enquanto a ideia central é a mensagem principal que o autor quer transmitir. Um texto pode defender a mesma ideia central a partir de diferentes pontos de vista. Por exemplo, dois textos podem defender a preservação do meio ambiente (mesma ideia central), mas um pode adotar um ponto de vista econômico (focando nos custos de desastres naturais) e o outro, um ponto de vista social (focando na qualidade de vida das futuras gerações).

— Argumentação

A argumentação é o processo pelo qual o autor apresenta e desenvolve suas ideias com o intuito de convencer ou persuadir o leitor. Em um texto argumentativo, a argumentação é fundamental para a construção de um raciocínio lógico e coeso que sustente a tese ou ponto de vista do autor. Ela se faz presente em diferentes tipos de textos, especialmente nos dissertativos, artigos de opinião, editoriais e ensaios, mas também pode ser encontrada de maneira indireta em textos literários e expositivos.

A qualidade da argumentação está diretamente ligada à clareza, à consistência e à relevância dos argumentos apresentados, além da capacidade do autor de antecipar e refutar possíveis contra-argumentos. Ao analisar a argumentação de um texto, é importante observar como o autor organiza suas ideias, quais recursos utiliza para justificar suas posições e de que maneira ele tenta influenciar o leitor.

– Estrutura da Argumentação

A argumentação em um texto dissertativo-argumentativo, por exemplo, costuma seguir uma estrutura lógica que inclui:

1. Tese: A tese é a ideia central que o autor pretende defender. Ela costuma ser apresentada logo no início do texto, frequentemente na introdução. A tese delimita o ponto de vista do autor sobre o tema e orienta toda a argumentação subsequente.

2. Argumentos: São as justificativas que sustentam a tese. Podem ser de vários tipos, como argumentos baseados em fatos, estatísticas, opiniões de especialistas, experiências concretas ou raciocínios lógicos. O autor utiliza esses argumentos para demonstrar a validade de sua tese e persuadir o leitor.

3. Contra-argumentos e Refutação: Muitas vezes, para fortalecer sua argumentação, o autor antecipa e responde a possíveis objeções ao seu ponto de vista. A refutação é uma estratégia eficaz que demonstra que o autor considerou outras perspectivas, mas que tem razões para desconsiderá-las ou contestá-las.

4. Conclusão: Na conclusão, o autor retoma a tese inicial e resume os principais pontos da argumentação, reforçando seu ponto de vista e buscando deixar uma impressão duradoura no leitor.

– Tipos de Argumentos

A argumentação pode utilizar diferentes tipos de argumentos, dependendo do objetivo do autor e do contexto do texto. Entre os principais tipos, podemos destacar:

1. Argumento de autoridade: Baseia-se na citação de especialistas ou de instituições renomadas para reforçar a tese. Esse tipo de argumento busca emprestar credibilidade à posição defendida.

Exemplo: "Segundo a Organização Mundial da Saúde (OMS), uma alimentação equilibrada pode reduzir em até 80% o risco de doenças crônicas, como diabetes e hipertensão."

2. Argumento de exemplificação: Utiliza exemplos concretos para ilustrar e validar o ponto de vista defendido. Esses exemplos podem ser tirados de situações cotidianas, casos históricos ou experimentos.

Exemplo: "Em países como a Suécia e a Finlândia, onde o sistema educacional é baseado na valorização dos professores, os índices de desenvolvimento humano são superiores à média global."

3. Argumento lógico (ou dedutivo): É baseado em um raciocínio lógico que estabelece uma relação de causa e efeito, levando o leitor a aceitar a conclusão apresentada. Esse tipo de argumento pode ser dedutivo (parte de uma premissa geral para uma conclusão específica) ou indutivo (parte de exemplos específicos para uma conclusão geral).

Exemplo dedutivo: "Todos os seres humanos são mortais. Sócrates é um ser humano. Logo, Sócrates é mortal."

Exemplo indutivo: "Diversos estudos demonstram que o uso excessivo de telas prejudica a visão. Portanto, o uso prolongado de celulares e computadores também pode afetar negativamente a saúde ocular."

4. Argumento emocional (ou patético): Apela aos sentimentos do leitor, utilizando a emoção como meio de convencimento. Este tipo de argumento pode despertar empatia, compaixão, medo ou revolta no leitor, dependendo da maneira como é apresentado.

Exemplo: "Milhares de crianças morrem de fome todos os dias enquanto toneladas de alimentos são desperdiçadas em países desenvolvidos. É inaceitável que, em pleno século XXI, ainda enfrentemos essa realidade."

5. Argumento de comparação ou analogia: Compara situações semelhantes para fortalecer o ponto de vista do autor. A comparação pode ser entre eventos, fenômenos ou comportamentos para mostrar que a lógica aplicada a uma situação também se aplica à outra.

Exemplo: "Assim como o cigarro foi amplamente aceito durante décadas, até que seus malefícios para a saúde fossem comprovados, o consumo excessivo de açúcar hoje deve ser visto com mais cautela, já que estudos indicam seus efeitos nocivos a longo prazo."

– Coesão e Coerência na Argumentação

A eficácia da argumentação depende também da coesão e coerência no desenvolvimento das ideias. Coesão refere-se aos mecanismos linguísticos que conectam as diferentes partes do texto, como pronomes, conjunções e advérbios. Estes elementos garantem que o texto flua de maneira lógica e fácil de ser seguido.

Exemplo de conectivos importantes:

- Para adicionar informações: "além disso", "também", "ademais".
- Para contrastar ideias: "no entanto", "por outro lado", "todavia".
- Para concluir: "portanto", "assim", "logo".

Já a coerência diz respeito à harmonia entre as ideias, ou seja, à lógica interna do texto. Um texto coerente apresenta uma relação clara entre a tese, os argumentos e a conclusão. A falta de coerência pode fazer com que o leitor perca o fio do raciocínio ou não aceite a argumentação como válida.

– Exemplos Práticos de Argumentação

- Texto Argumentativo (Artigo de Opinião): Em um artigo que defenda a legalização da educação domiciliar no Brasil, a tese pode ser que essa prática oferece mais liberdade educacional para os pais e permite uma personalização do ensino. Os argumentos poderiam incluir exemplos de países onde a educação domiciliar é bem-sucedida, dados sobre o desempenho acadêmico de crianças educadas em casa e opiniões de especialistas. O autor também pode refutar os argumentos de que essa modalidade de ensino prejudica a socialização das crianças, citando estudos que mostram o contrário.

- Texto Literário: Em obras literárias, a argumentação pode ser mais sutil, mas ainda está presente. No romance "Capitães da Areia", de Jorge Amado, embora a narrativa siga a vida de crianças abandonadas nas ruas de Salvador, a estrutura do texto e a escolha dos eventos apresentados constroem uma crítica implícita à desigualdade social e à falta de políticas públicas eficazes. A argumentação é feita de maneira indireta, por meio das experiências dos personagens e do ambiente descrito.

– Análise Crítica da Argumentação

Para analisar criticamente a argumentação de um texto, é importante que o leitor:

1. Avalie a pertinência dos argumentos: Os argumentos são válidos e relevantes para sustentar a tese? Estão bem fundamentados?

2. Verifique a solidez da lógica: O raciocínio seguido pelo autor é coerente? Há falácias argumentativas que enfraquecem a posição defendida?

3. Observe a diversidade de fontes: O autor utiliza diferentes tipos de argumentos (fatos, opiniões, dados) para fortalecer sua tese, ou a argumentação é unilateral e pouco fundamentada?

4. Considere os contra-argumentos: O autor reconhece e refuta pontos de vista contrários? Isso fortalece ou enfraquece a defesa da tese?

— Elementos de Coesão

Os elementos de coesão são os recursos linguísticos que garantem a conexão e a fluidez entre as diferentes partes de um texto. Eles são essenciais para que o leitor compreenda como as ideias estão relacionadas e para que o discurso seja entendido de forma clara e lógica. Em termos práticos, a coesão se refere à capacidade de manter as frases e parágrafos interligados, criando uma progressão lógica que permite ao leitor seguir o raciocínio do autor sem perder o fio condutor.

A coesão textual pode ser alcançada por meio de diversos mecanismos, como o uso de conectivos, pronomes, elipses e sinônimos, que evitam repetições desnecessárias e facilitam a transição entre as ideias. Em textos argumentativos e dissertativos, esses elementos desempenham um papel fundamental na organização e no desenvolvimento da argumentação.

– Tipos de Coesão

Os principais tipos de coesão podem ser divididos em coesão referencial, coesão sequencial e coesão lexical. Cada um deles envolve diferentes estratégias que contribuem para a unidade e a clareza do texto.

1. Coesão Referencial

A coesão referencial ocorre quando um elemento do texto remete a outro já mencionado, garantindo que as ideias sejam retomadas ou antecipadas sem a necessidade de repetição direta. Isso pode ser feito por meio de pronomes, advérbios ou outras expressões que retomam conceitos, pessoas ou objetos mencionados anteriormente.

ENGENHARIA DE SOFTWARE ENGENHARIA DE REQUISITOS: PRINCIPAIS TÉCNICAS DE ELICITAÇÃO DE REQUISITOS. CASOS DE USO E USER STORIES. GESTÃO DE BACKLOG. PRODUTO MÍNIMO VIÁVEL (MVP). GESTÃO DE DÍVIDA TÉCNICA

Engenharia De Requisitos

Principais Técnicas De Elicitação De Requisitos

A elicitação de requisitos é o processo de coleta e definição das necessidades dos stakeholders para o desenvolvimento de um sistema. As principais técnicas incluem:

- **Entrevistas**: Conversas estruturadas ou semiestruturadas com os stakeholders apara entender suas necessidades e expectativas.
- **Workshops:** Reuniões colaborativas onde os stakeholders e a equipe de desenvolvimento discutem requisitos e priorizam funcionalidades.
- **Brainstorming:** Sessão de ideias para levantar requisitos de maneira criativa e abrangente.
- **Questionários**: Formulários estruturados para coletar informações de um grande número de pessoas.
- **Observação direta:** Acompanhamento das atividades dos usuários para entender como o sistema deve apoiar suas tarefas.
- **Análise de documentos:** Avaliação de relatórios, fluxogramas e outros documentos para identificar requisitos implícitos.
- **Protótipos**: Criação de modelos visuais do sistema para validar requisitos e obter feedback dos stakeholders.
- **Casos de uso:** Representação de interações entre usuários e o sistema para definir cenários de uso práticos.

Casos De Uso E User Stories

Casos de uso e user stories são abordagens para documentar e compreender os requisitos do sistema. Embora tenham propósitos semelhantes, eles diferem em formato e aplicação.

1. Casos de uso:

- São descrições detalhadas das interações entre o usuário e o sistema;
- Contêm elementos como atores, fluxos de eventos, condições e exceções;
- São representados por diagramas UML e descrições textuais.

2. User stories:

- São histórias curtas que descrevem a necessidade do usuário de forma simples;
- Utilizam o formato: “Como [tipo de usuário], eu quero [objetivo] para que [benefício].”;
- São amplamente utilizadas em metodologias ágeis e priorizadas no backlog.

Gestão De Backlog

A gestão de backlog é essencial para organizar e priorizar tarefas em projetos de software. Um backlog bem gerenciado melhora a produtividade e entrega de valor ao cliente.

Principais aspectos:

- **Definição do backlog:** Lista priorizada de requisitos, funcionalidades e melhorias do produto.
- **Priorização:** Técnicas como MoSCoW, WSJF (Weighted Shortest Job First) e ROI ajudam na definição da ordem de implementação.
- **Refinamento do backlog:** Revisão constante para esclarecer, dividir e ajustar as tarefas conforme necessário.
- **Papel do Product Owner:** Responsável por manter o backlog atualizado e alinhado com os objetivos do negócio.
- **Integração com metodologias ágeis:** Scrum e Kanban utilizam o backlog para planejar sprints e gerenciar fluxos de trabalho.

Produto Mínimo Viável (Mvp)

O MVP (“Minimum Viable Product”) é uma abordagem que busca validar ideias com o menor esforço possível.

Principais características:

- **Definição:** Uma versão básica do produto que oferece valor ao usuário e permite a coleta de feedback.
- **Objetivo:** Testar hipóteses e aprender sobre o mercado com investimentos reduzidos.

Ciclo de desenvolvimento:

- Construir: Criar a versão mais simples do produto.
- Medir: Coletar dados sobre a usabilidade e aceitação.
- Aprender: Ajustar e evoluir o produto com base no feedback.

Exemplos: Protótipos interativos, landing pages, versões beta de software.

Gestão De Dívida Técnica

Dívida técnica refere-se a decisões que aceleram o desenvolvimento no curto prazo, mas que geram custos adicionais no futuro.

Principais pontos:

Tipos de dívida técnica:

▪ Intencional: Decisões conscientes para entregar rapidamente;

▪ Involuntária: Causada por falta de conhecimento ou falhas no processo.

Impactos:

▪ Aumento de custos de manutenção;

▪ Redução da qualidade do código;

▪ Dificuldade na implementação de novas funcionalidades.

Gestão eficaz:

▪ Monitoramento contínuo com ferramentas como SonarQube;

▪ Refatoração frequente para reduzir complexidade;

▪ Definição de prazos para a resolução da dívida;

▪ Cultura de qualidade e boas práticas de desenvolvimento.

ANÁLISE E PROJETO DE SOFTWARE: MODELAGEM E DESIGN UTILIZANDO UML. PADRÕES DE PROJETO (DESIGN PATTERNS). PROGRAMAÇÃO ORIENTADA A OBJETOS (CONCEITOS GERAIS)

Análise E Projeto De Software
Modelagem E Design Utilizando Uml

A UML (Unified Modeling Language) é uma linguagem padronizada para modelagem de sistemas de software, utilizada para representar graficamente a estrutura e o comportamento do sistema.

Principais diagramas UML:

▪ **Diagrama de Casos de Uso**: Representa as interações entre os atores (usuários) e o sistema.

▪ **Diagrama de Classes**: Exibe a estrutura do sistema, com classes, atributos, métodos e relações entre elas.

▪ **Diagrama de Sequência**: Ilustra a interação entre objetos ao longo do tempo.

▪ **Diagrama de Atividade**: Mostra o fluxo de execução de processos no sistema.

▪ **Diagrama de Estado**: Representa os estados de um objeto e as transições entre eles.

▪ **Diagrama de Componentes**: Mostra a estrutura física dos componentes de software
e suas dependências.

▪ **Diagrama de Implantação**: Representa a distribuição física dos componentes em servidores e dispositivos.

A UML facilita a comunicação entre desenvolvedores, clientes e demais stakeholders, proporcionando uma visão clara do sistema antes da implementação.

Padrões De Projeto (Design Patterns)

Os padrões de projeto são soluções reutilizáveis para problemas comuns no desenvolvimento de software. Eles são classificados em três categorias principais:

Padrões Criacionais: Auxiliam na criação de objetos, aumentando a flexibilidade e reutilização de código.

▪ Singleton: Garante que uma classe tenha apenas uma instância.

▪ Factory Method: Permite a criação de objetos sem especificar sua classe exata.

▪ Abstract Factory: Fornece uma interface para criação de famílias de objetos relacionados.

Padrões Estruturais: Definem como classes e objetos são compostos para formar estruturas maiores.

▪Adapter: Permite que interfaces incompatíveis trabalhem juntas.

▪ Composite: Organiza objetos em uma estrutura hierárquica de árvore.

▪ Decorator: Adiciona funcionalidades a um objeto dinamicamente.

Padrões Comportamentais: Gerenciam interações entre objetos e a distribuição de responsabilidades.

▪ Observer: Define uma dependência entre objetos para notificação automática de mudanças.

▪ Strategy: Permite alterar dinamicamente o comportamento de um objeto.

▪ Command: Encapsula uma solicitação como um objeto, permitindo seu
armazenamento e execução posterior.

Os padrões de projeto facilitam a manutenção e expansão do software, promovendo boas práticas de engenharia de software.

Programação Orientada A Objetos (Conceitos Gerais)

A Programação Orientada a Objetos (POO) é um paradigma baseado na organização do código em torno de objetos, que representam entidades do mundo real. Seus principais conceitos são:

▪ **Encapsulamento**: Restringe o acesso direto aos atributos de um objeto, protegendo os dados e garantindo integridade.

▪ **Herança**: Permite que uma classe reutilize e estenda atributos e métodos de outra classe.

▪ **Polimorfismo**: Habilita objetos de diferentes classes a serem tratados de maneira uniforme por meio de interfaces ou classes base.

▪ **Abstração**: Destaca os aspectos essenciais de um objeto, ignorando detalhes desnecessários.

A POO melhora a reutilização de código, a modularização e a manutenção do software, tornando-o mais flexível e escalável.

QUALIDADE DE SOFTWARE: ANÁLISE ESTÁTICA DE CÓDIGO. TESTES (UNITÁRIOS, DE INTEGRAÇÃO, NÃO FUNCIONAIS). MOCKING E STUBS. REVISÃO DE CÓDIGO E PROGRAMAÇÃO EM PAR

Qualidade De Software

Análise Estática De Código

A análise estática de código é uma técnica utilizada para avaliar a qualidade do software sem a necessidade de executá-lo. Essa abordagem permite identificar vulnerabilidades, padrões de código incorretos e melhorias na estrutura do programa.

Principais ferramentas e técnicas:

- **Linters**: Ferramentas como ESLint (JavaScript), Pylint (Python) e Checkstyle (Java) ajudam a detectar erros de estilo e padrões de código.
- **Analisadores de Código**: Ferramentas como SonarQube e CodeClimate avaliam métricas de qualidade e detectam vulnerabilidades.
- **Regras de Complexidade**: Identifica trechos de código com complexidade excessiva, facilitando a manutenção e compreensão do software.
- **Detecção de Vulnerabilidades**: Identifica falhas de segurança, como SQL Injection e Cross-Site Scripting (XSS).

Benefícios Da Análise Estática Do Código

- Identifica problemas antes da execução do software;
- Reduz erros em produção;
- Melhora a legibilidade e manutenção do código.

Testes Unitários

Testes unitários são testes automatizados focados em verificar o comportamento de componentes individuais do sistema, como funções, métodos e classes.

Principais características:

- Testam partes isoladas do código;
- Executam rapidamente e possuem baixo custo de manutenção;
- Utilizam frameworks como JUnit (Java), NUnit (C#) e PyTest (Python).

Boas práticas:

- Seguir o padrão AAA (Arrange, Act, Assert).
- Escrever testes independentes e determinísticos.
- Cobrir casos de sucesso e falha.

Benefícios dos testes unitários

- Reduz erros durante o desenvolvimento;
- Facilita a refatoração segura do código;
- Aumenta a confiança na estabilidade do software.

Testes De Integração

Os testes de integração verificam a interação entre diferentes módulos ou componentes do sistema para garantir que funcionem corretamente quando combinados.

Principais características:

- Validam a comunicação entre APIs, bancos de dados e outros serviços;
- Podem ser escritos usando frameworks como TestNG (Java), Mocha (JavaScript) e Robot Framework (Python);
- Utilizam técnicas como testes em camadas e mocks para isolar dependências.

Benefícios dos testes de integração:

- Detectam falhas na interação entre componentes;
- Garantem a compatibilidade entre diferentes partes do sistema;
- Melhoram a confiabilidade da aplicação.

Testes Não Funcionais

Os testes não funcionais avaliam atributos do sistema como desempenho, segurança, usabilidade e escalabilidade.

Principais tipos:

- **Testes de desempenho**: Avaliam tempo de resposta e carga do sistema (exemplo: JMeter, Gatling);
- **Testes de segurança**: Identificam vulnerabilidades e falhas de segurança (exemplo: OWASP ZAP, Burp Suite);
- **Testes de usabilidade**: Avaliam a experiência do usuário e a interface do sistema;
- **Testes de compatibilidade**: Verificam a execução em diferentes dispositivos, navegadores e sistemas operacionais.

Benefícios dos testes não funcionais

- Melhoram a estabilidade e segurança do software;
- Garantem um melhor desempenho em ambientes reais;
- Reduzem riscos de falhas críticas.

Mocking E Stubs

Mocking e Stubs são técnicas usadas em testes automatizados para simular comportamentos de componentes externos ao código testado.

Mocks:

- Simulam objetos reais e permitem a verificação de chamadas de método;
- Utilizados para testar interações entre componentes.

Exemplo de ferramentas: Mockito (Java), Moq (C#), unittest.mock (Python).

Stubs:

- Retornam respostas pré-definidas para chamadas específicas;
- Usados para reduzir a dependência de sistemas externos em testes.

Benefícios dos mockin e stubs:

- Permitem testar módulos independentemente de serviços externos;
- Melhoram a confiabilidade e performance dos testes;
- Reduzem custos de execução de testes.

Revisão De Código E Programação Em Par

A revisão de código e a programação em par são práticas que ajudam a garantir a qualidade do software através da colaboração entre desenvolvedores.

Revisão de Código:

- Envolve a análise do código por outro desenvolvedor ou equipe;
- Ferramentas como GitHub Pull Requests, Gerrit e Crucible são usadas para facilitar esse processo.
- Objetivos: detectar erros, melhorar a manutenção e compartilhar conhecimento.

Programação em Par (Pair Programming):

- Dois desenvolvedores trabalham juntos no mesmo código: um escreve e o outro revisa;
- Promove maior qualidade, melhor design e aprendizado contínuo.

Benefícios Da Revisão De Código E Programação Em Par:

- Reduzem erros e retrabalho;
- Melhoram a compreensão e padronização do código;
- Aumentam a colaboração e eficiência da equipe.

INFRAESTRUTURA COMO CÓDIGO (IAC) (INFRA): CONCEITOS E FERRAMENTAS (ANSIBLE, TERRAFORM, SHELLSCRIPT). AUTOMAÇÃO DE PROVISIONAMENTO DE AMBIENTES

Infraestrutura Como Código (IAC)

Conceitos E Ferramentas: Ansible

O **Ansible** é uma ferramenta de automação de infraestrutura que permite gerenciar configuração, provisionamento e implantação de aplicações de forma eficiente. Ele utiliza uma abordagem baseada em **declaratividade**, onde os estados desejados são descritos em arquivos YAML (playbooks).

Principais conceitos:

- **Idempotência**: Garante que a execução repetida dos scripts resulte no mesmo estado desejado
- **Arquitetura sem agentes**: Diferente de outras ferramentas, o Ansible se conecta a máquinas remotas via SSH, sem necessidade de instalar um agente.
- **Módulos reutilizáveis**: Oferece uma ampla biblioteca de módulos para diversas tarefas (gerenciamento de pacotes, configuração de serviços, etc.).

Exemplo de playbook Ansible:

```
- hosts: servidores
  tasks:
    - name: Instalar Apache
      apt:
        name: apache2
        state: present
```

Benefícios do Ansible:

- Simplicidade e facilidade de aprendizado.
- Integração com diversas plataformas.
- Redução de erros humanos na configuração de ambientes.

Conceitos E Ferramentas: Terraform

O **Terraform** é uma ferramenta de infraestrutura como código (IaC) usada para provisionamento e gerenciamento de recursos em nuvem. Ele utiliza uma abordagem **declarativa** por meio do HashiCorp Configuration Language (**HCL**).

Principais conceitos:

- **State Management**: Mantém um estado da infraestrutura para rastrear e gerenciar mudanças.
- **Providers**: APIs que permitem interagir com provedores como AWS, Azure, GCP, entre outros.
- **Plan, Apply e Destroy**: Processo de execução para criar, modificar e remover recursos de forma segura.

Exemplo de configuração Terraform:

```
provider “aws” {
  region = “us-east-1”
}

resource “aws_instance” “web” {
  ami           = “ami-12345678”
  instance_type = “t2.micro”
}
```

Benefícios do Terraform:

- Infraestrutura reproduzível e escalável;
- Automatiza o provisionamento em múltiplos provedores;
- Facilita a auditoria e controle de versões da infraestrutura.

Conceitos E Ferramentas: Shellscript

O **ShellScript** é uma linguagem de script utilizada para automação de tarefas em sistemas Unix/Linux. Ele é muito usado na gestão de infraestrutura para instalação, configuração e execução de comandos automatizados.

Principais conceitos:

- **Interpretação de comandos**: Sequência de comandos executados automaticamente.
- **Uso de variáveis**: Permite armazenar e reutilizar valores.
- **Estruturas de controle**: Uso de laços, condicionais e funções para lógicas mais complexas.

Exemplo de ShellScript para instalar o Apache:

```
#!/bin/bash
sudo apt update -y
sudo apt install apache2 -y
sudo systemctl start apache2
```

Benefícios do ShellScript:

- Simplicidade e compatibilidade com sistemas Unix/Linux;
- Permite automação de tarefas repetitivas;
- Integração com outras ferramentas de IaC.

VISA AVALIAR A HABILIDADE DO(A) CANDIDATO(A) EM ENTENDER A ESTRUTURA LÓGICA DAS RELAÇÕES ARBITRÁRIAS ENTRE PESSOAS, LUGARES, COISAS, EVENTOS FICTÍCIOS; DEDUZIR NOVAS INFORMAÇÕES DAS RELAÇÕES FORNECIDAS E AVALIAR AS CONDIÇÕES USADAS PARA ESTABELECER A ESTRUTURA DAQUELAS RELAÇÕES. ESTRUTURAS LÓGICAS, LÓGICAS DE ARGUMENTAÇÃO, DIAGRAMAS LÓGICOS

A capacidade de estabelecer e interpretar relações lógicas entre diferentes elementos é uma habilidade essencial para o desenvolvimento do pensamento analítico. Essa competência permite ao indivíduo organizar informações, identificar padrões e criar conexões relevantes, mesmo diante de conceitos abstratos ou situações hipotéticas. Ao dominar esse campo, é possível analisar premissas, avaliar sua consistência e extrair conclusões fundamentadas, promovendo uma compreensão mais profunda e decisões mais acertadas. Essa habilidade é indispensável na resolução de problemas complexos e no enfrentamento de desafios que exigem clareza e raciocínio estruturado.

A seguir, exploraremos os principais conteúdos que ajudam a aprimorar essa competência:

LÓGICA PROPOSICIONAL

Um predicado é uma sentença que contém um número limitado de variáveis e se torna uma proposição quando são dados valores às variáveis matemáticas e propriedades quaisquer a outros tipos.

Um predicado, de modo geral, indica uma relação entre objetos de uma afirmação ou contexto.

Considerando o que se conhece da língua portuguesa e, intuitivamente, predicados dão qualidade aos sujeitos, relacionam os sujeitos e relacionam os sujeitos aos objetos.

Para tal, são usados os conectivos lógicos $\neg, \Rightarrow, \longrightarrow, \wedge, \vee$, mais objetos, predicados, variáveis e quantificadores.

Os objetos podem ser concretos, abstratos ou fictícios, únicos (atômicos) ou compostos.

Logo, é um tipo que pode ser desde uma peça sólida, um número complexo até uma afirmação criada para justificar um raciocínio e que não tenha existência real!

Os argumentos apresentam da lógica dos predicados dizem respeito, também, àqueles da lógica proposicional, mas adicionando as qualidades ao sujeito.

As palavras que relacionam os objetos são usadas como quantificadores, como um objeto está sobre outro, um é maior que o outro, a cor de um é diferente da cor do outro; e, com o uso dos conectivos, as sentenças ficam mais complexas.

Por exemplo, podemos escrever que um objeto é maior que outro e eles têm cores diferentes.

Somando as variáveis aos objetos com predicados, as variáveis definem e estabelecem fatos relativos aos objetos em um dado contexto.

Vamos examinar as características de argumentos e sentenças lógicas para adentrarmos no uso de quantificadores.

No livro Discurso do Método de René Descartes, encontramos a afirmação: "(1ª parte): "...a diversidade de nossas opiniões não provém do fato de serem uns mais racionais que outros, mas somente de conduzirmos nossos pensamentos por vias diversas e não considerarmos as mesmas coisas. Pois não é suficiente ter o espírito bom, o principal é aplicá-lo bem."

Cabe aqui, uma rápida revisão de conceitos, como o de **argumento**, que é a afirmação de que um grupo de proposições gera uma proposição final, que é consequência das primeiras. São ideias lógicas que se relacionam com o propósito de esclarecer pontos de pensamento, teorias, dúvidas.

Seguindo a ideia do princípio para o fim, a proposição é o início e o argumento o fim de uma explanação ou raciocínio, portanto essencial para um pensamento lógico.

A proposição ou sentença a é uma oração declarativa que poderá ser classificada somente em verdadeira ou falsa, com sentido completo, tem sujeito e predicado.

Por exemplo, e usando informações multidisciplinares, são proposições:

I – A água é uma molécula polar;

II – A membrana plasmática é lipoprotéica.

Observe que os exemplos acima seguem as condições essenciais que uma proposição deve seguir, i.e., dois axiomas fundamentais da lógica, [1] o princípio da não contradição e [2] o princípio do terceiro excluído, como já citado.

O princípio da não contradição afirma que uma proposição não ser verdadeira e falsa ao mesmo tempo.

O princípio do terceiro excluído afirma que toda proposição ou é verdadeira ou é falsa, jamais uma terceira opção.

Após essa pequena revisão de conceitos, que representaram os tipos de argumentos chamados válidos, vamos especificar os conceitos para construir argumento inválidos, falaciosos ou sofisma.

Proposições simples e compostas

Para se construir as premissas ou hipóteses em um argumento válido logicamente, as premissas têm extensão maior que a conclusão. A primeira premissa é chamada de maior é a mais abrangente, e a menor, a segunda, possui o sujeito da conclusão para o silogismo; e das conclusões, temos que:

I – De duas premissas negativas, nada se conclui;

II – De duas premissas afirmativas não pode haver conclusão negative;

III – A conclusão segue sempre a premissa mais fraca;

IV – De duas premissas particulares, nada se conclui.

As premissas funcionam como proposições e podem ser do tipo simples ou composta. As compostas são formadas por duas ou mais proposições simples interligadas por um "conectivo".

Uma proposição/premissa é toda oração declarativa que pode ser classificada em verdadeira ou falsa ou ainda, um conjunto de palavras ou símbolos que exprimem um pensamento de sentido completo.

Características de uma proposição:

I – Tem sujeito e predicado;

II – É declarativa (não é exclamativa nem interrogativa);

III – Tem um, e somente um, dos dois valores lógicos: ou é verdadeira ou é falsa.

É regida por princípios ou axiomas:

I – Princípio da não contradição: uma proposição não pode ser verdadeira e falsa ao mesmo tempo.

II – Princípio do terceiro excluído: toda proposição ou é verdadeira ou é falsa, isto é, verifica-se sempre um destes casos e nunca um terceiro.

Exemplos:

– A água é uma substância polar.

– A membrana plasmática é lipoprotéica.

– As premissas podem ser unidas via conectivos mostrados na tabela abaixo e já mostrado acima. São eles:

Proposição	Forma	Símbolo
Negação	Não	¬
Disjunção não exclusiva	ou	∨
Conjunção	e	^
Condicional	Se... então	→
Bicondicional	Se e somente se	↔

Tabelas verdade

As tabelas-verdade são ferramentas utilizadas para analisar as possíveis combinações de valores lógicos (verdadeiro ou falso) das proposições. Elas permitem compreender o comportamento lógico de operadores como negação, conjunção e disjunção, facilitando a verificação da validade de proposições compostas. Abaixo, apresentamos as tabelas-verdade para cada operador,

1. Negação

A partir de uma proposição *p* qualquer, pode-se construir outra, a negação de *p*, cujo símbolo é ¬*p*.

Exemplos:

A água é uma substância não polar.

A membrana plasmática é não lipoprotéica.

Tabela-verdade para *p* e ¬*p*.

p	¬*p*
V	F
F	V

Os símbolos lógicos para construção de proposições compostas são: ∧ (lê-se e) e ∨ (lê-se ou).

2. Conectivo ∧:

Colocando o conectivo ∧ entre duas proposições *p* e *q*, obtém-se uma nova proposição *p* ∧ *q*, denominada conjunção das sentenças.

Exemplos:

p: substâncias apolares atravessam diretamente a bicamada lipídica.

q: o aminoácido fenilalanina é apolar.

p ∧ *q*: substâncias apolares atravessam diretamente a bicamada lipídica e o aminoácido fenilalanina é apolar.

Tabela-verdade para a conjunção

Axioma: a conjunção é verdadeira se, e somente se, ambas as proposições são verdadeiras; se ao menos uma delas for falsa, a conjunção é falsa.

p	*q*	*p* ∧ *q*
V	V	V
V	F	F
F	V	F
F	F	F

3. Conectivo ∨:

Colocando o conectivo ∨ entre duas proposições *p* e *q*, obtém-se uma nova proposição *p* ∨ *q*, denominada disjunção das sentenças.

Exemplos:

p: substâncias apolares atravessam diretamente a bicamada lipídica.

q: substâncias polares usam receptores proteicos para atravessar a bicamada lipídica.

p ∨ *q*: substâncias apolares atravessam diretamente a bicamada lipídica ou substâncias polares usam receptores proteicos para atravessar a bicamada lipídica.

Tabela-verdade para a disjunção

Axioma: a disjunção é verdadeira se ao menos das duas proposições for verdadeira; se ambas forem falsas, então a disjunção é falsa.

p	*q*	*p* ∨ *q*
V	V	V
V	F	V
F	V	V
F	F	F

Símbolos lógicos para sentenças condicionais são: se ...então... (símbolo →); ...se, e somente se, ... (símbolo ↔).

4. Condicional →

O condicional → colocado entre *p* e *q*, obtém-se uma nova proposição *p* → *q*, que se lê :se p então q, '*p* é condição necessária para *q*' e '*q* é condição suficiente para *p*'

p é chamada antecedente e *q* é chamada de consequente.

Exemplos:

p: o colesterol é apolar.

q: o colesterol penetra a bicamada lipídica.

p → *q*: se o colesterol é apolar, então o colesterol penetra a bicamada lipídica.

Tabela-verdade para a condicional →

Axioma: o condicional *p* → *q* é falsa somente quando *p* é verdadeira e *q* é falsa, caso contrário, *p* → *q* é verdadeira.

p	***q***	***p → q***
V	V	V
V	F	F
F	V	V
F	F	V

5. Bicondicional ↔

O bicondicional ↔ colocado entre *p* e *q*, obtém-se uma nova proposição *p* ↔ *q* que se lê :*p* se, somente se, *q*, '*q* é condição necessária e suficiente para *p*' e 'se *p*, então *q* e reciprocamente'

Exemplos:

p: o colesterol é uma substância apolar.

q: o colesterol não é solúvel em água.

p ↔ *q*: o colesterol é uma substância apolar se, e somente se, o colesterol não é solúvel em água.

Tabela-verdade para a bicondicional ↔

Axioma: o bicondicional ↔ é verdadeiro somente quando p e q são ambas verdadeiras ou ambas são falsas.

p	***q***	***p ↔ q***
V	V	V
V	F	F
F	V	F
F	F	V

Tautologia, Contradição e Contingência

As proposições compostas podem ser classificadas de acordo com o seu valor lógico final, considerando todas as possíveis combinações de valores lógicos das proposições simples que as compõem. Essa classificação é fundamental para entender a validade de argumentos lógicos:

Tautologia

Uma tautologia é uma proposição composta cujo valor lógico final é sempre verdadeiro, independentemente dos valores das proposições simples que a compõem. Em outras palavras, não importa se as proposições simples são verdadeiras ou falsas; a proposição composta será sempre verdadeira. Tautologias ajudam a validar raciocínios. Se uma proposição complexa é tautológica, então o argumento que a utiliza é logicamente consistente e sempre válido.

Exemplo: A proposição "p ou não-p" (ou p v ~p) é uma tautologia porque, seja qual for o valor de p (verdadeiro ou falso), a proposição composta sempre terá um resultado verdadeiro. Isso reflete o Princípio do Terceiro Excluído, onde algo deve ser verdadeiro ou falso, sem meio-termo.

Contradição

Uma contradição é uma proposição composta que tem seu valor lógico final sempre falso, independentemente dos valores lógicos das proposições que a compõem. Assim, qualquer que seja o valor das proposições simples, o resultado será falso. Identificar contradições em um argumento é essencial para determinar inconsistências lógicas. Quando uma proposição leva a uma contradição, isso significa que o argumento em questão não pode ser verdadeiro.

Exemplo: A proposição "p e não-p" (ou p ^ ~p) é uma contradição, pois uma proposição não pode ser verdadeira e falsa ao mesmo tempo. Esse exemplo reflete o Princípio da Não Contradição, que diz que uma proposição não pode ser simultaneamente verdadeira e falsa.

Contingência

Uma contingência é uma proposição composta cujo valor lógico final pode ser tanto verdadeiro quanto falso, dependendo dos valores das proposições simples que a compõem. Diferentemente das tautologias e contradições, que são invariavelmente verdadeiras ou falsas, as contingências refletem casos em que o valor lógico não é absoluto e depende das circunstâncias. Identificar contradições em um argumento é essencial para determinar inconsistências lógicas. Quando uma proposição leva a uma contradição, isso significa que o argumento em questão não pode ser verdadeiro.

Exemplo: A proposição "se p então q" (ou p → q) é uma contingência, pois pode ser verdadeira ou falsa dependendo dos valores de p e q. Caso p seja verdadeiro e q seja falso, a proposição composta será falsa. Em qualquer outra combinação, a proposição será verdadeira.

Exemplo:

4. (CESPE) Um estudante de direito, com o objetivo de sistematizar o seu estudo, criou sua própria legenda, na qual identificava, por letras, algumas afirmações relevantes quanto à disciplina estudada e as vinculava por meio de sentenças (proposições). No seu vocabulário particular constava, por exemplo:

P: Cometeu o crime A.

Q: Cometeu o crime B.

R: Será punido, obrigatoriamente, com a pena de reclusão no regime fechado.

S: Poderá optar pelo pagamento de fiança.

Ao revisar seus escritos, o estudante, apesar de não recordar qual era o crime B, lembrou que ele era inafiançável.Tendo como referência essa situação hipotética, julgue o item que se segue.

A sentença (P→Q)↔((~Q)→(~P)) será sempre verdadeira, independentemente das valorações de P e Q como verdadeiras ou falsas.

( ) CERTO
( ) ERRADO

Resolução:

Temos a sentença (P→Q)↔((~Q)→(~P)).

Sabemos que (~Q)→(~P) é equivalente a P→Q, entao podemos substituir:

P→Q ↔ P→Q

Considerando P→Q = A, temos:

A ↔ A

Uma bicondicional (↔) é verdadeira quando ambos os lados têm o mesmo valor lógico.

Como ambos os lados são A, eles sempre terão o mesmo valor.

Logo a sentença é sempre verdadeira, independentemente dos valores de P e Q.

Resposta: Certo.

Equivalências

O nome equivalência deriva de igualdade ou coisas que se equivalem, e dentro de coisas, entenda-se também, raciocínio.

Em termos de lógica, se duas proposições possuem o mesmo resultado para suas tabelas-verdade, elas são ditas equivalentes e se escreve *p=q*. o caso mais simples se verifica na negação da negação de uma proposição, i.e., ~(~p). como exemplo veja a tabela-verdade abaixo.

p	***q***	***p ∨ q***	***~(p ∨ q)***	***~p ∧ ~q***
V	V	V	**F**	**F**
V	F	V	**F**	**F**
F	V	V	**F**	**F**
F	F	F	**V**	**V**

Logo, *~(p ∨ q)* e *~p ∧ ~q*, são proposições equivalentes.

Temos, dentro do raciocínio lógico as equivalências básicas cujas deduções são lógicas e diretas:

I – *p ∧ p = p*
II – p ∨ p = p
III – p ∧ q = q ∧ p
IV – p ∨ q = q ∨ p

Para mostrar a lógica simples das sentenças acima, pense que, para (I), se algo escrevermos que *estudar matemática é bom* e que *estudar matemática é bom*, logicamente, deduzimos que *estudar matemática é bom*!!

Leis de Morgan

Dentro das equivalências, existem as equivalências ou leis de De Morgan, que se referem às negações das proposições do tipo *negação da conjunção* e sua equivalência com a disjunção, assim como *negação da disjunção* e sua equivalência com a conjunção, como segue:

¬(p ∧ q) = ¬p ∨ ¬q
¬(p ∨ q) = ¬p ∧ ¬q

Implicações

Uma proposição P(p,q,r,...) implica logicamente ou apenas implica uma proposição Q(p,q,r,...) se Q(p,q,r,...) é verdadeira (V) todas as vezes que P(p,q,r,...) é verdadeira (V), ou seja, a proposição P implica a proposição Q, quando a condicional P → Q for uma tautologia.

Representamos a implicação com o símbolo "⇒", simbolicamente temos:

P(p,q,r,...) ⇒ Q(p,q,r,...).

A não ocorrência de VF na tabela verdade de P → Q, ou ainda que o valor lógico da condicional P → Q será sempre V, ou então que P → Q é uma tautologia.

Observação: Os símbolos "→" e "⇒" são completamente distintos. O primeiro ("→") representa a condicional, que é um conectivo. O segundo ("⇒") representa a relação de implicação lógica que pode ou não existir entre duas proposições.

Exemplo:

A tabela verdade da condicional (p ^ q) → (p ↔ q) será:

p	q	p ^ q	p ↔ q	(p ^ q) → (p ↔ q)
V	V	V	V	V
V	F	F	F	V
F	V	F	F	V
F	F	F	V	V

Portanto, (p ^ q) → (p ↔ q) é uma tautologia, por isso (p ^ q) ⇒ (p ↔q).

Em particular:

– Toda proposição implica uma Tautologia: p ⇒ p v ~p

p	p v ~p
V	V
F	V

– Somente uma contradição implica uma contradição: p ^ ~p ⇒ p v ~p → p ^ ~p

p	~p	p ^ ~p	p v ~p → p ^ ~p
V	F	F	F
F	V	F	F