



FRANCA - SP

PREFEITURA MUNICIPAL DE FRANCA - SÃO PAULO

ANALISTA DE SISTEMAS

EDITAL DE ABERTURA DO CONCURSO
PÚBLICO Nº 05/2026

CÓD: OP-069JH-26
7908403596294

Língua Portuguesa

1. Compreensão e interpretação de textos verbais; Identificação de informações explícitas e implícitas; Tema, finalidade, tese, argumentos e efeitos de sentido.....	7
2. Coesão e coerência textual.....	7
3. Relações de sentido entre palavras, expressões, frases e parágrafos; Sinonímia, antonímia, denotação e conotação	8
4. Tipologia textual e gêneros textuais.....	9
5. Ortografia oficial.....	10
6. Acentuação gráfica.....	11
7. Pontuação.....	12
8. Classes de palavras e seus empregos no texto.....	19
9. Flexão nominal e verbal.....	25
10. Concordância nominal e verbal.....	28
11. Regência nominal e verbal.....	30
12. Crase.....	31
13. Colocação pronominal.....	31
14. Reescrita de frases e períodos, com manutenção do sentido e da correção gramatical.....	32

Raciocínio Lógico

1. Estruturas lógicas. Proposições. Conectivos lógicos. Tabelas-verdade. Negação de proposições simples e compostas. Equivalências lógicas. Implicação lógica.....	45
2. Argumentos válidos e inválidos.....	50
3. Diagramas lógicos.....	54
4. Sequências numéricas, alfabéticas e figurais. Padrões e regularidades.....	54
5. Raciocínio dedutivo e indutivo.....	60
6. Resolução de problemas lógicos.....	63

Conhecimentos Específicos Analista de Sistemas

1. Engenharia de software: conceitos, processos, ciclo de vida de sistemas, levantamento, análise, especificação e validação de requisitos. Análise e projeto de sistemas. Modelagem de sistemas. UML: casos de uso, classes, sequência, atividades e estados. Análise e projeto orientados a objetos.....	69
2. Algoritmos e estruturas de dados. Lógica de programação.....	72
3. Programação orientada a objetos.....	74
4. Desenvolvimento de sistemas web. Arquitetura de aplicações. APIs, serviços web, REST, JSON, XML e integração de sistemas.....	75
5. Banco de dados relacionais: modelagem entidade-relacionamento, normalização, integridade referencial, transações, índices, visões, procedimentos armazenados e otimização de consultas. Linguagem SQL. Noções de bancos de dados não relacionais.....	78
6. Sistemas operacionais Windows e Linux.....	89
7. Redes de computadores: conceitos básicos, protocolos, TCP/IP, DNS, DHCP, HTTP, HTTPS, redes locais, redes sem fio e noções de infraestrutura.....	94

ÍNDICE

8. Noções de computação em nuvem, virtualização e containers.....	103
9. Segurança da informação: confidencialidade, integridade, disponibilidade, autenticação, autorização, controle de acesso, criptografia, certificados digitais, firewall, backup, recuperação de dados, vulnerabilidades, malware, phishing e boas práticas de segurança.....	105
10. Proteção de dados pessoais e Lei Geral de Proteção de Dados Pessoais, Lei nº 13.709/2018.....	110
11. Ética, sigilo, responsabilidade profissional e segurança no tratamento de informações no serviço público.....	124
12. Testes de software: testes unitários, integrados, funcionais, regressivos, desempenho, homologação e aceite. Qualidade de software. Documentação de sistemas, manuais, rotinas operacionais e documentação técnica.	126
13. Implantação, manutenção, sustentação e evolução de sistemas. DevOps, controle de versionamento e integração contínua: uso de repositórios de código, controle de versões, pipelines de integração e entrega contínua, automação de testes, implantação automatizada, reversão de implantação, monitoramento e boas práticas de colaboração entre desenvolvimento e operação. Desempenho, escalabilidade, monitoramento e otimização de sistemas.....	129
14. Suporte técnico a usuários. Gestão de incidentes, problemas, mudanças e requisições de serviço.....	133
15. Noções de governança e gestão de serviços de TI.....	136
16. Métodos ágeis: Scrum e Kanban. Gestão de projetos de software	143

LÍNGUA PORTUGUESA

COMPREENSÃO E INTERPRETAÇÃO DE TEXTOS VERBAIS; IDENTIFICAÇÃO DE INFORMAÇÕES EXPLÍCITAS E IMPLÍCITAS; TEMA, FINALIDADE, TESE, ARGUMENTOS E EFEITOS DE SENTIDO

Compreender e interpretar textos é essencial para que o objetivo de comunicação seja alcançado satisfatoriamente. Com isso, é importante saber diferenciar os dois conceitos. Vale lembrar que o texto pode ser verbal ou não-verbal, desde que tenha um sentido completo.

A **compreensão** se relaciona ao entendimento de um texto e de sua proposta comunicativa, decodificando a mensagem explícita. Só depois de compreender o texto que é possível fazer a sua interpretação.

A **interpretação** são as conclusões que chegamos a partir do conteúdo do texto, isto é, ela se encontra para além daquilo que está escrito ou mostrado. Assim, podemos dizer que a interpretação é subjetiva, contando com o conhecimento prévio e do repertório do leitor.

Dessa maneira, para compreender e interpretar bem um texto, é necessário fazer a decodificação de códigos linguísticos e/ou visuais, isto é, identificar figuras de linguagem, reconhecer o sentido de conjunções e preposições, por exemplo, bem como identificar expressões, gestos e cores quando se trata de imagens.

Dicas práticas

- Faça um resumo (pode ser uma palavra, uma frase, um conceito) sobre o assunto e os argumentos apresentados em cada parágrafo, tentando traçar a linha de raciocínio do texto. Se possível, adicione também pensamentos e inferências próprias às anotações.
- Tenha sempre um dicionário ou uma ferramenta de busca por perto, para poder procurar o significado de palavras desconhecidas.
- **Fique atento aos detalhes oferecidos pelo texto:** dados, fonte de referências e datas.
- 4. Sublinhe as informações importantes, separando fatos de opiniões.
- **Perceba o enunciado das questões. De um modo geral, questões que esperam compreensão do texto aparecem com as seguintes expressões:** o autor afirma/sugere que...; segundo o texto...; de acordo com o autor... Já as questões que esperam interpretação do texto aparecem com as seguintes expressões: conclui-se do texto que...; o texto permite deduzir que...; qual é a intenção do autor quando afirma que...

COESÃO E COERÊNCIA TEXTUAL

A coerência e a coesão são essenciais na escrita e na interpretação de textos. Ambos se referem à relação adequada entre os componentes do texto, de modo que são independentes entre si. Isso quer dizer que um texto pode estar coeso, porém incoerente, e vice-versa.

Enquanto a coesão tem foco nas questões gramaticais, ou seja, ligação entre palavras, frases e parágrafos, a coerência diz respeito ao conteúdo, isto é, uma sequência lógica entre as ideias.

► Coesão

A coesão textual ocorre, normalmente, por meio do uso de **conectivos** (preposições, conjunções, advérbios). Ela pode ser obtida a partir da **anáfora** (retoma um componente) e da **catáfora** (antecipa um componente).

Confira, então, as principais regras que garantem a coesão textual:

REGRA	CARACTERÍSTICAS	EXEMPLOS
REFERÊNCIA	<ul style="list-style-type: none"> – Pessoal (uso de pronomes pessoais ou possessivos) <ul style="list-style-type: none"> – anafórica – Demonstrativa (uso de pronomes demonstrativos e advérbios) – catafórica – Comparativa (uso de comparações por semelhanças) 	<p>João e Maria são crianças. <i>Eles</i> são irmãos.</p> <p>Fiz todas as tarefas, exceto <u>esta</u>: colonização africana.</p> <p>Mais um ano <u>igual</u> aos outros...</p>



AMOSTRA

SUBSTITUIÇÃO	– Substituição de um termo por outro, para evitar repetição	Maria está triste. A <u>menina</u> está cansada de ficar em casa.
ELIPSE	– Omissão de um termo	No quarto, apenas quatro ou cinco convidados. (omissão do verbo “haver”)
CONJUNÇÃO	– Conexão entre duas orações, estabelecendo relação entre elas	Eu queria ir ao cinema, <u>mas</u> estamos de quarentena.
COESÃO LEXICAL	– Utilização de sinônimos, hiperônimos, nomes genéricos ou palavras que possuem sentido aproximado e pertencente a um mesmo grupo lexical.	A minha <u>casa</u> é clara. Os <u>quartos</u> , a <u>sala</u> e a <u>cozinha</u> têm janelas grandes.

► **Coerência**

Nesse caso, é importante conferir se a mensagem e a conexão de ideias fazem sentido, e seguem uma linha clara de raciocínio. Existem alguns conceitos básicos que ajudam a garantir a coerência. Veja quais são os principais princípios para um texto coerente:

- **Princípio da não contradição:** não deve haver ideias contraditórias em diferentes partes do texto.
- **Princípio da não tautologia:** a ideia não deve estar redundante, ainda que seja expressa com palavras diferentes.
- **Princípio da relevância:** as ideias devem se relacionar entre si, não sendo fragmentadas nem sem propósito para a argumentação.
- **Princípio da continuidade temática:** é preciso que o assunto tenha um seguimento em relação ao assunto tratado.
- **Princípio da progressão semântica:** inserir informações novas, que sejam ordenadas de maneira adequada em relação à progressão de ideias.

Para atender a todos os princípios, alguns fatores são recomendáveis para garantir a coerência textual, como amplo **conhecimento de mundo**, isto é, a bagagem de informações que adquirimos ao longo da vida; **inferências** acerca do conhecimento de mundo do leitor; e **informatividade**, ou seja, conhecimentos ricos, interessantes e pouco previsíveis.

RELAÇÕES DE SENTIDO ENTRE PALAVRAS, EXPRESSÕES, FRASES E PARÁGRAFOS; SINONÍMIA, ANTONÍMIA, DENOTAÇÃO E CONOTAÇÃO

Este é um estudo da **semântica**, que pretende classificar os sentidos das palavras, as suas relações de sentido entre si. Conheça as principais relações e suas características:

► **Sinonímia e antonímia**

As palavras **sinônimas** são aquelas que apresentam significado semelhante, estabelecendo relação de proximidade.

Ex.: inteligente <—> esperto

Já as palavras **antônimas** são aquelas que apresentam significados opostos, estabelecendo uma relação de contrariedade.

Ex.: forte <—> fraco

► **Parônimos e homônimos**

As palavras **parônimas** são aquelas que possuem grafia e pronúncia semelhantes, porém com significados distintos.

Ex.: cumprimento (saudação) X comprimento (extensão); tráfico (trânsito) X tráfico (comércio ilegal).

As palavras **homônimas** são aquelas que possuem a mesma grafia e pronúncia, porém têm significados diferentes.

Ex.: rio (verbo “rir”) X rio (curso d’água); manga (blusa) X manga (fruta).

As palavras **homófonas** são aquelas que possuem a mesma pronúncia, mas com escrita e significado diferentes.

Ex.: cem (numeral) X sem (falta); concerto (arrumar) X concerto (musical).

As palavras **homógrafas** são aquelas que possuem escrita igual, porém som e significado diferentes.

Ex.: colher (talher) X colher (verbo); acerto (substantivo) X acerto (verbo).

► **Polissemia e monosssemia**

As palavras **polissêmicas** são aquelas que podem apresentar mais de um significado, a depender do contexto em que ocorre a frase.

Ex.: cabeça (parte do corpo humano; líder de um grupo).



RACIOCÍNIO LÓGICO

ESTRUTURAS LÓGICAS. PROPOSIÇÕES. CONECTIVOS LÓGICOS. TABELAS-VERDADE. NEGAÇÃO DE PROPOSIÇÕES SIMPLES E COMPOSTAS. EQUIVALÊNCIAS LÓGICAS. IMPLICAÇÃO LÓGICA

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 $\neg, \Rightarrow, \rightarrow, \wedge, \vee$, 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:

- De duas premissas negativas, nada se conclui;
- De duas premissas afirmativas não pode haver conclusão negativa;
- A conclusão segue sempre a premissa mais fraca;
- 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

- Tem sujeito e predicado;
- É declarativa (não é exclamativa nem interrogativa);
- **Tem um, e somente um, dos dois valores lógicos:** ou é verdadeira ou é falsa.

AMOSTRA

É regida por princípios ou axiomas:

- **Princípio da não contradição:** uma proposição não pode ser verdadeira e falsa ao mesmo tempo.
- **Princípio do terceiro excluído:** toda proposição ou é verdadeira ou é falsa, isto é, verifica-se sempre um destes casos e nunca um terceiro.
- **Princípio da Identidade:** uma proposição é idêntica a si mesma. Em termos simples: $p \equiv p$

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	\neg
Disjunção não exclusiva	ou	\vee
Conjunção	e	\wedge
Condicional	Se... então	\rightarrow
Bicondicional	Se e somente se	\leftrightarrow

► 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,

Negação

A partir de uma proposição p qualquer, pode-se construir outra, a negação de p , cujo símbolo é $\neg p$.

Exemplos:

- A água é uma substância não polar.
- A membrana plasmática é não lipoprotéica.

Tabela-verdade para p e $\neg p$.

p	$\neg p$
V	F
F	V

Os símbolos lógicos para construção de proposições compostas são: \wedge (lê-se e) e \vee (lê-se ou).

Conectivo e

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

Exemplos:

- p : substâncias apolares atravessam diretamente a bicamada lipídica.
- q : o aminoácido fenilalanina é apolar.
- $p \wedge q$: 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	$p \wedge q$
V	V	V
V	F	F
F	V	F
F	F	F

Conectivo ou

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

Exemplos:

- p : substâncias apolares atravessam diretamente a bicamada lipídica.
- q : substâncias polares usam receptores proteicos para atravessar a bicamada lipídica.
- $p \vee q$: 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	$p \vee q$
V	V	V
V	F	V
F	V	V
F	F	F

Símbolos lógicos para sentenças condicionais são: se ...então... (símbolo \rightarrow); ...se, e somente se, ... (símbolo \leftrightarrow).

CONHECIMENTOS ESPECÍFICOS

ENGENHARIA DE SOFTWARE: CONCEITOS, PROCESSOS, CICLO DE VIDA DE SISTEMAS, LEVANTAMENTO, ANÁLISE, ESPECIFICAÇÃO E VALIDAÇÃO DE REQUISITOS. ANÁLISE E PROJETO DE SISTEMAS. MODELAGEM DE SISTEMAS. UML: CASOS DE USO, CLASSES, SEQUÊNCIA, ATIVIDADES E ESTADOS. ANÁLISE E PROJETO ORIENTADOS A OBJETOS

FUNDAMENTOS DA ENGENHARIA DE SOFTWARE E PROCESSOS DE DESENVOLVIMENTO

► Conceito de Engenharia de Software

A Engenharia de Software é a área da computação que estuda métodos, técnicas, processos e ferramentas para desenvolver, manter e evoluir sistemas de software com qualidade. Seu objetivo é organizar o desenvolvimento de sistemas de forma planejada, controlada e alinhada às necessidades dos usuários e das organizações.

Um software não deve ser entendido apenas como código. Ele envolve requisitos, regras de negócio, modelos, documentação, interfaces, dados, integrações, testes, implantação e manutenção. Por isso, a Engenharia de Software busca reduzir improvisos e tornar o desenvolvimento mais previsível, compreensível e sustentável.

Sem práticas de engenharia, sistemas podem se tornar difíceis de manter, inconsistentes, caros, inseguros ou desalinhados com aquilo que os usuários realmente precisam.

► Objetivos da Engenharia de Software

A Engenharia de Software procura garantir que o sistema desenvolvido atenda a determinados objetivos. Entre os principais, estão:

- Compreender corretamente o problema a ser resolvido
- Definir requisitos claros e verificáveis
- Projetar soluções adequadas ao contexto
- Organizar o desenvolvimento em etapas controladas
- Facilitar manutenção, evolução e documentação do sistema

Esses objetivos mostram que a área não se limita à programação. Antes da implementação, é necessário compreender o problema, especificar necessidades, modelar soluções e validar se o caminho escolhido faz sentido.

► Processos de software

Um processo de software é um conjunto organizado de atividades usadas para desenvolver e manter sistemas. Ele define etapas, responsabilidades, produtos gerados e critérios de acompanhamento. Um processo pode ser mais formal ou mais flexível, dependendo do projeto, da equipe e da organização.

De modo geral, processos de software envolvem atividades como levantamento de requisitos, análise, projeto, implementação, validação, implantação e manutenção. Essas atividades podem ocorrer de forma sequencial, iterativa ou incremental, conforme o modelo adotado.

O processo ajuda a equipe a saber o que deve ser feito, quando deve ser feito e com quais critérios.

► Características de um processo bem definido

Um processo bem definido precisa ser compreensível, adaptável e útil. Ele não deve existir apenas como burocracia, mas como instrumento para melhorar a organização do trabalho.

Algumas características importantes são:

- Clareza nas etapas e responsabilidades
- Critérios para aprovação e validação de entregas
- Documentação suficiente para orientar decisões
- Controle das mudanças ao longo do desenvolvimento
- Comunicação adequada entre usuários, analistas e desenvolvedores

Um processo inadequado pode gerar atraso e excesso de formalismo. Já a ausência de processo pode gerar desorganização e retrabalho.

► Importância da sistematização do desenvolvimento

A sistematização do desenvolvimento é importante porque sistemas de software costumam envolver muitas decisões, pessoas, requisitos e restrições. Quando essas decisões não são registradas ou organizadas, a equipe pode perder entendimento sobre o funcionamento do sistema.

A Engenharia de Software permite transformar necessidades em soluções técnicas de forma mais controlada. Isso melhora a qualidade do produto, facilita a comunicação entre os envolvidos e aumenta a capacidade de manutenção futura.

CICLO DE VIDA DE SISTEMAS E ENGENHARIA DE REQUISITOS

► Conceito de ciclo de vida de sistemas

O ciclo de vida de um sistema representa as etapas pelas quais um software passa desde sua concepção até sua desativação ou substituição. Ele inclui o nascimento da ideia, o estudo das necessidades, a análise, o projeto, a construção, a validação, a implantação, a manutenção e a evolução.

AMOSTRA

Essa visão é importante porque um sistema não termina quando é colocado em uso. Após a implantação, surgem correções, melhorias, mudanças de regras, adaptações tecnológicas e novas demandas dos usuários.

O ciclo de vida ajuda a compreender que o software é um produto dinâmico, que precisa ser acompanhado ao longo do tempo.

► Fases do desenvolvimento de software

Embora existam diferentes modelos de ciclo de vida, algumas fases aparecem com frequência. A primeira é a identificação da necessidade. Depois ocorre o levantamento de requisitos, seguido da análise e da especificação. Em seguida, são definidos o projeto da solução, a implementação e a validação. Por fim, o sistema é implantado e mantido.

Essas fases não precisam ocorrer sempre de forma rígida. Em muitos projetos, há revisões e retornos entre elas. Ainda assim, a separação conceitual ajuda a entender o papel de cada atividade.

► Levantamento de requisitos

O levantamento de requisitos consiste em descobrir o que os usuários e a organização esperam do sistema. Essa etapa pode envolver entrevistas, reuniões, observação de processos, análise de documentos, questionários e estudo de sistemas existentes.

O objetivo é identificar necessidades reais, restrições, regras de negócio, problemas atuais e expectativas sobre a solução.

Um bom levantamento deve buscar respostas para perguntas como:

- Quem usará o sistema?
- Que atividades o sistema deverá apoiar?
- Quais informações serão registradas ou consultadas?
- Quais regras precisam ser respeitadas?
- Que dificuldades existem no processo atual?

Essa etapa é fundamental para evitar que a solução seja construída com base em suposições.

► Análise e especificação de requisitos

Após o levantamento, os requisitos precisam ser analisados. A análise busca organizar, complementar, priorizar e resolver conflitos entre necessidades. Muitas vezes, usuários diferentes têm expectativas distintas, e cabe à equipe compreender o que é essencial, possível e coerente.

A especificação de requisitos registra formalmente o que o sistema deverá fazer e quais condições deverá atender. Essa documentação pode incluir descrições textuais, regras de negócio, fluxos, casos de uso, protótipos e modelos.

Um requisito bem especificado deve ser claro, necessário, verificável e sem ambiguidade.

► Validação e gerenciamento de requisitos

A validação de requisitos verifica se aquilo que foi documentado representa corretamente a necessidade dos usuários. Não basta escrever requisitos; é preciso confirmar se eles fazem sentido e se foram compreendidos por todos.

O gerenciamento de requisitos acompanha mudanças durante o projeto. Como necessidades podem evoluir, é necessário controlar alterações, avaliar impactos e manter coerência entre requisitos, modelos e solução construída.

ANÁLISE E PROJETO DE SISTEMAS

► Objetivos da análise de sistemas

A análise de sistemas busca compreender o problema e definir o que o sistema deverá realizar. Ela se concentra no domínio do problema, nos processos envolvidos, nas informações manipuladas e nas necessidades dos usuários.

Durante a análise, o foco ainda não está nos detalhes técnicos de implementação. O objetivo principal é entender o funcionamento esperado do sistema e as regras que orientarão seu comportamento.

A análise responde principalmente à pergunta: o que o sistema precisa fazer?

► Objetivos do projeto de sistemas

O projeto de sistemas transforma a análise em uma solução estruturada. Nessa etapa, são definidas decisões sobre organização interna, componentes, classes, interfaces, dados, interações e responsabilidades dos elementos do sistema.

Enquanto a análise está mais próxima do problema, o projeto está mais próximo da solução. Ele responde à pergunta: como o sistema será organizado para atender aos requisitos?

Um bom projeto facilita implementação, manutenção e evolução.

► Transformação de requisitos em soluções

A passagem dos requisitos para o projeto é uma das etapas mais importantes do desenvolvimento. Um requisito como “permitir o cadastro de usuários” precisa ser transformado em telas, regras de validação, classes, operações, armazenamento e mensagens de retorno.

Essa transformação exige interpretação técnica e cuidado para manter coerência com a necessidade original. Se a análise for mal feita, o projeto pode resolver o problema errado. Se o projeto for mal estruturado, a implementação pode se tornar difícil de manter.

► Visões funcional e estrutural do sistema

A análise e o projeto costumam observar o sistema por diferentes visões. A visão funcional mostra o que o sistema faz, quais funcionalidades oferece e como os usuários interagem com ele. A visão estrutural mostra quais elementos compõem o sistema e como eles se relacionam.

Essas visões se complementam. Um sistema precisa ser compreendido tanto pelo comportamento esperado quanto pela estrutura que sustentará esse comportamento.