COM BASE NO ÚLTIMO EDITAL

apostilas opção

PETROBRAS

PETRÓLEO BRASILEIRO S.A

ANALISTA DE SISTEMAS
ENGENHARIA DE SOFTWARE

- Língua Portuguesa
- Matemática
- Conhecimentos Específicos - Bloco I
- Conhecimentos Específicos - Bloco II
- Conhecimentos Específicos - Bloco III

BÔNUS
CURSO ON-LINE

- PORTUGUÊS
- INFORMÁTICA

⚠ AVISO IMPORTANTE:

Este é um Material de Demonstração

Este arquivo representa uma prévia exclusiva da apostila.

Aqui, você poderá conferir algumas páginas selecionadas para conhecer de perto a qualidade, o formato e a proposta pedagógica do nosso conteúdo. Lembramos que este não é o material completo.

🎯 POR QUE INVESTIR NA APOSTILA COMPLETA?

- ✕ Conteúdo totalmente alinhado ao edital.
- ✕ Teoria clara, objetiva e sempre atualizada.
- ✕ Dicas práticas, quadros de resumo e linguagem descomplicada.
- ✕ Questões gabaritadas
- ✕ Bônus especiais que otimizam seus estudos.

Aproveite a oportunidade de intensificar sua preparação com um material completo e focado na sua aprovação:
Acesse agora: www.apostilasopcao.com.br

Disponível nas versões impressa e digital, com envio imediato!

Estudar com o material certo faz toda a diferença na sua jornada até a APROVAÇÃO.

PETROBRAS

PETRÓLEO BRASILEIRO S.A

ANALISTA DE SISTEMAS - ENGENHARIA DE SOFTWARE

COM BASE NO EDITAL N.º 1 – PETROBRAS/ PSP RH 2021, DE 15 DE DEZEMBRO DE 2021

CÓD: OP-055FV-26
7908403588473

Língua Portuguesa

Língua Inglesa

Conhecimentos Específicos - Bloco I
Analista de Sistemas - Engenharia de Software

Conhecimentos Específicos - Bloco II

Conhecimentos Específicos - Bloco III

LÍNGUA PORTUGUESA

COMPREENSÃO E INTERPRETAÇÃO DE TEXTOS DE GÊNEROS VARIADOS

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:

▪ 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.

▪ **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.

▪ **Seleção de Argumentos:** Nos textos argumentativos, os exemplos, dados e fatos apresentados pelo autor refletem o ponto de vista defendido. Textos favoráveis a uma

LÍNGUA INGLESA

COMPREENSÃO DE TEXTOS ESCRITOS EM LÍNGUA INGLESA

Reading Comprehension

Interpretar textos pode ser algo trabalhoso, dependendo do assunto, ou da forma como é abordado. Tem as questões sobre o texto. Mas, quando o texto é em outra língua? Tudo pode ser mais assustador.

Se o leitor manter a calma, e se embasar nas estratégias do Inglês Instrumental e ter certeza que ninguém é cem por cento leigo em nada, tudo pode ficar mais claro.

Vejamos o que é e quais são suas estratégias de leitura:

Inglês Instrumental

Também conhecido como Inglês para Fins Específicos - ESP, o Inglês Instrumental fundamenta-se no treinamento instrumental dessa língua. Tem como objetivo essencial proporcionar ao aluno, em curto prazo, a capacidade de ler e compreender aquilo que for de extrema importância e fundamental para que este possa desempenhar a atividade de leitura em uma área específica.

Estratégias de leitura

- **Skimming:** trata-se de uma estratégia onde o leitor vai buscar a ideia geral do texto através de uma leitura rápida, sem apegar-se a ideias mínimas ou específicas, para dizer sobre o que o texto trata.
- **Scanning:** através do scanning, o leitor busca ideias específicas no texto. Isso ocorre pela leitura do texto à procura de um detalhe específico. Praticamos o scanning diariamente para encontrarmos um número na lista telefônica, selecionar um e-mail para ler, etc.
- **Cognatos:** são palavras idênticas ou parecidas entre duas línguas e que possuem o mesmo significado, como a palavra "vírus" é escrita igualmente em português e inglês, a única diferença é que em português a palavra recebe acentuação. Porém, é preciso atentar para os chamados falsos cognatos, ou seja, palavras que são escritas igual ou parecidas, mas com o significado diferente, como "evaluation", que pode ser confundida com "evolução" onde na verdade, significa "avaliação".
- **Inferência contextual:** o leitor lança mão da inferência, ou seja, ele tenta adivinhar ou sugerir o assunto tratado pelo texto, e durante a leitura ele pode confirmar ou descartar suas hipóteses.
- **Reconhecimento de gêneros textuais:** são tipo de textos que se caracterizam por organização, estrutura gramatical, vocabulário específico e contexto social em que ocorrem. Dependendo das marcas textuais, podemos distinguir uma poesia de uma receita culinária, por exemplo.
- **Informação não-verbal:** é toda informação dada através de figuras, gráficos, tabelas, mapas, etc. A informação não-verbal deve ser considerada como parte da informação ou ideia que o texto deseja transmitir.
- **Palavras-chave:** são fundamentais para a compreensão do texto, pois se trata de palavras relacionadas à área e ao assunto abordado pelo texto. São de fácil compreensão, pois, geralmente, aparecem repetidamente no texto e é possível obter sua ideia através do contexto.
- **Grupos nominais:** formados por um núcleo (substantivo) e um ou mais modificadores (adjetivos ou substantivos). Na língua inglesa o modificador aparece antes do núcleo, diferente da língua portuguesa.
- **Afixos:** são prefixos e/ou sufixos adicionados a uma raiz, que modifica o significado da palavra. Assim, conhecendo o significado de cada afixo pode-se compreender mais facilmente uma palavra composta por um prefixo ou sufixo.
- **Conhecimento prévio:** para compreender um texto, o leitor depende do conhecimento que ele já tem e está armazenado em sua memória. É a partir desse conhecimento que o leitor terá o entendimento do assunto tratado no texto e assimilará novas informações. Trata-se de um recurso essencial para o leitor formular hipóteses e inferências a respeito do significado do texto.

O leitor tem, portanto, um papel ativo no processo de leitura e compreensão de textos, pois é ele que estabelecerá as relações entre aquele conteúdo do texto e os conhecimentos de mundo que ele carrega consigo. Ou mesmo, será ele que poderá agregar mais profundidade ao conteúdo do texto a partir de sua capacidade de buscar mais conhecimentos acerca dos assuntos que o texto traz e sugere.

Não se esqueça que saber interpretar textos em inglês é muito importante para ter melhor acesso aos conteúdos escritos fora do país, ou para fazer provas de vestibular ou concursos.

ITENS GRAMATICAIS RELEVANTES PARA O ENTENDIMENTO DOS SENTIDOS DOS TEXTOS

Dentre os muitos tópicos gramaticais da língua inglesa, alguns se fazem primordiais para a compreensão textual e a contextualização da comunicação no idioma. Os tempos verbais são as principais gramáticas a serem estudadas para uma melhor compreensão do idioma por completo. Ao realizar a interpretação de um texto, deve-se levar o tempo verbal em consideração para que se possa contextualizar o momento ao qual a fala se refere. Confira a seguir.

Simple present

O *simple present* ou o presente simples é marcado por dois verbos auxiliares específicos DO e DOES. A conjugação verbal no tempo presente da língua inglesa é simples e se divide entre grupos de sujeitos. No infinitivo, ou seja, quando terminados em "ar", "er", "ir" no português, o verbo leva "to" em inglês, veja a seguir.

- Comer – to *eat*
- Beber – to *drink*
- Andar – to *walk*

Todos os verbos no presente mantêm uma conjugação básica, muito mais simples que a do português para cada sujeito. Basta retirar o "to" do infinitivo para serem conjugados com os sujeitos *I, you, we, they* e *you* (plural). Veja:

- I **eat** – Eu como
- You **eat** – Você come/ Tu comes
- We **eat** – Nós comemos
- They **eat** – Eles comem
- You **eat** – Vocês comem/ Vós comeis

No caso dos pronomes na terceira pessoa (*he*, *she* e *it*), acrescenta-se ao verbo o s conjuga-los adequadamente no tempo presente; para saber quando usar casa partícula, é necessário atentar-se ao final de cada verbo. Veja:

- She speaks Spanish.
- My brother enjoys watching movies.
- Anne visits her family on weekends

A grande maioria dos verbos recebem a terminação em s no inglês, em especial os terminados em sons consonantais de p, t, k ou f ou sons vogais. Mas encontramos algumas exceções também em que devemos acresentar es ou ies ao final do verbo, no caso de verbos terminados em y, em ch, em sh, em x, em s ou em z.

Em verbos a terminação consoante + y, acrescenta-se o "*ies*". Confira alguns exemplos de verbos que se encaixam nesta regra.

- *To study – She* studies *math*. (Ela estuda matemática)
- *To try – He* tries *to practice sports.* (Ele tenta praticar esportes)
- *To fry – John* fries *potatoes in oil.* (John frita batatas no óleo)
- *To copy – Lucy* copies *the text.* (Lucy copia o texto)
- *To reply – He* replies *with a text.* (Ele responde com uma mensagem)

Há, porém, uma exceção para a regra do "y". Em verbos que seguem a ordem de consoante, vogal e consoante (cvc) em sua terminação, acrescenta-se apenas o "*s*". Confira:

- *To play - She* plays *the guitar*. (Ela toca violão)
- *To stay* – It **stays** there (Fica lá)
- *To enjoy* – He **enjoys** playing the piano. (Ele gosta de tocar o violão)

Verbos terminados em ch, sh, s, z ou x, terminam "*es*". Observe:

- *To touch – He* touches *his nose*. (Ele toca seu nariz)
- *To press – Mary* presses *the button.* (Maria aperta o botão)
- *To buzz – The noise* buzzes *across the room.* (O barulho zumbe pela sala)
- *To crash* – The bus **crashes** against the wall (O ônibus bate contra o muro)
- *To fix* – The man **fixes** the sink. (O homem conserta a pia)

Observe que apenas no caso dos pronomes em terceira pessoa (he, she, it), o verbo se modificou. Nos demais sujeitos o verbo mantem sua forma original do infinitivo.

Há ainda o uso dos verbos auxiliares DO e DOES em frases negativas e interrogativas no presente simples do inglês. E, assim como a conjugação verbal, os auxiliares são divididos em dois grupos de acordo com os sujeitos:

- **DO** para *I, You, We, They e You* (plural).
- **DOES** para *He, She e It.*

Na negativa, o verbo auxiliar do ou does é somado ao not (não), podendo sofrer uma contração, comum da linguagem informal.

- Do not = **don't**
- Does not = **doesn't**

Sendo assim, no presente acrescentam-se estes auxiliares ao modo negativo para formular uma frase negativa. O verbo que o segue, porém, retorna ao seu estado primário (infinitivo sem "to") em todos os casos quando as frases estão na forma negativa. Veja:

- *You* do not enjoy *this song. / You* don't enjoy *this song*

  (Você não gosta desta canção)
- *She* does not understand *English / She* doesn't understand *English.*

  (Ela não entende inglês)

Em frases interrogativas os verbos auxiliares do presente são postos no início da frase e o verbo retorna para seu estado infinitivo sem o "to". Confira:

- Do *you enjoy watching TV?* (Você gosta de assistir TV?)
- Do *Anna and Joe undertand the text?* (Anna e John entendem o texto?)
- Does *she work at a store?* (Ela trabalha em uma loja?)
- Does *Matt speak Mandarin?* (Matt fala mandarim?)

E assim formamos as bases das estruturas do tempo presente na língua inglesa.

Simple past

O passado simples no inglês segue uma estrutura ainda mais simplificada do que o próprio presente simples. O auxiliar DID é responsável por formular frases negativas e interrogativas. E os verbos são divididos entre verbos regulares e irregulares.

CONHECIMENTOS ESPECÍFICOS - BLOCO I

ENGENHARIA DE SOFTWARE: MODELOS DE CICLO DE VIDA DE SOFTWARE; METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE; ARQUITETURA DE SOFTWARE; CONCEITOS E TÉCNICAS DO PROJETO DE SOFTWARE; PROCESSOS E PRÁTICAS DE DESENVOLVIMENTO DE SOFTWARE; PROCESSO INTERATIVO E INCREMENTAL; PRÁTICAS ÁGEIS DE DESENVOLVIMENTO DE SOFTWARE; GERENCIAMENTO DE CICLO DE VIDA DE APLICAÇÕES; DESENVOLVIMENTO ORIENTADO POR COMPORTAMENTO (BDD); DESENVOLVIMENTO GUIADO POR TESTES (TDD); INTEGRAÇÃO CONTÍNUA; DIAGRAMA ENTIDADE RELACIONAMENTO (ER); NOTAÇÃO BPMN; CONCEITOS E FERRAMENTAS DE DEVOPS; TÉCNICAS DE INTEGRAÇÃO E IMPLANTAÇÃO CONTÍNUA DE CÓDIGO (CI/CD)

Ciclos de Vida e Metodologias de Desenvolvimento

O **Ciclo de Vida de Desenvolvimento de Software (SDLC)** é o framework que define as etapas (Requisitos, Design, Implementação, Testes, Implantação e Manutenção) pelas quais um sistema passa.

Modelos de Ciclo de Vida de Software

Modelo em Cascata (Waterfall)

É o modelo tradicional, linear e sequencial. Uma fase só se inicia quando a anterior está 100% concluída.

- **Vantagem:** Disciplina, planejamento claro e marcos de entrega definidos.
- **Desvantagem:** Alta resistência a mudanças. O cliente só vê o software no final, o que eleva o risco de o produto não atender às expectativas.

Modelo em Espiral (Spiral)

Focado primordialmente em **Análise de Risco**. O projeto é desenvolvido em ciclos (voltas na espiral), onde cada volta passa por planejamento, análise de risco, engenharia e avaliação.

- **Ideal para:** Projetos de missão crítica, grandes e complexos onde o risco de falha é inaceitável.

Processo Iterativo e Incremental

É a base do desenvolvimento moderno. O projeto é dividido em pequenas partes (incrementos). A cada ciclo (iteração), uma versão funcional do software é produzida e refinada.

- **Iterativo:** O software é refinado através de ciclos de feedback.
- **Incremental:** Novas funcionalidades são adicionadas a cada ciclo.

Metodologias e Práticas Ágeis

As metodologias ágeis surgiram para combater a rigidez do modelo Cascata, priorizando a adaptabilidade e a entrega de valor contínua.

Scrum

É o framework ágil mais utilizado. Baseia-se em ciclos de trabalho fixos chamados **Sprints** (geralmente de 2 a 4 semanas).

- **Papéis:** Product Owner (define o valor), Scrum Master (facilita o processo) e Time de Desenvolvimento.
- **Artefatos:** Product Backlog, Sprint Backlog e o Incremento (software pronto).
- **Eventos:** Sprint Planning, Daily Scrum, Sprint Review e Sprint Retrospective.

Kanban

Diferente do Scrum, o Kanban não possui ciclos fixos (Sprints). Ele foca na **visualização do fluxo de trabalho**.

WIP (Work In Progress): O conceito crucial do Kanban é limitar a quantidade de tarefas em andamento para evitar gargalos e garantir que o time finalize o que começou antes de puxar novas demandas.

Programação Extrema (XP)

Foca na excelência técnica e na qualidade do código. Introduz práticas como **Pair Programming** (programação em par), integração contínua e a simplicidade do design.

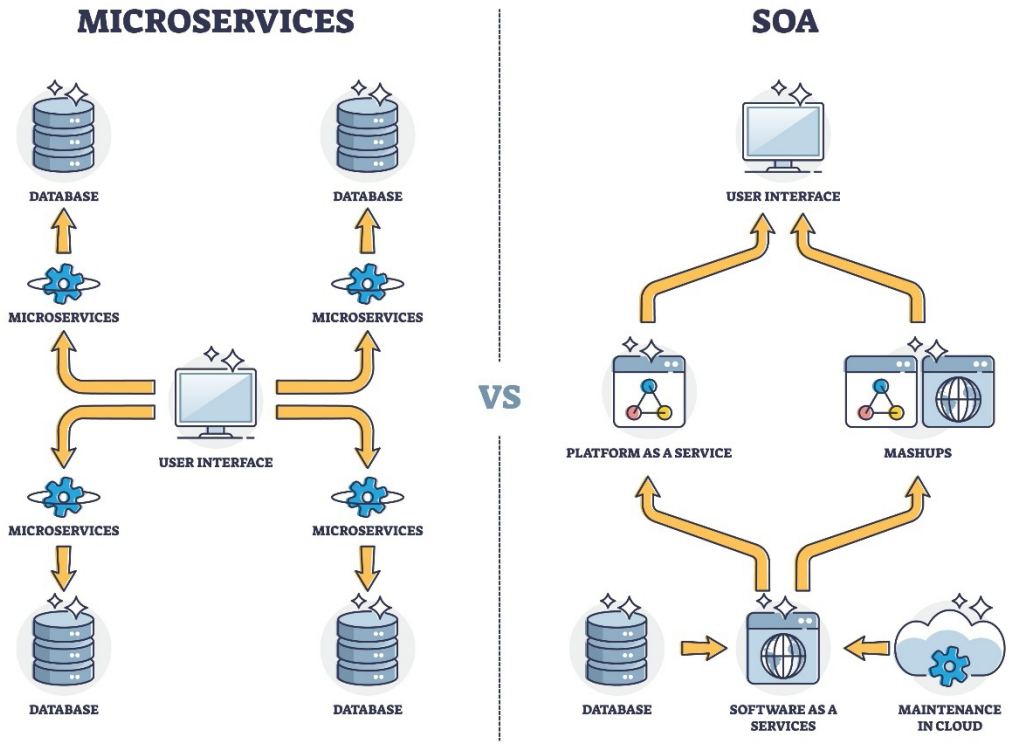
Gerenciamento de Ciclo de Vida de Aplicações (ALM)

O ALM é o "guarda-chuva" que engloba todo o processo. Enquanto o SDLC foca no desenvolvimento, o ALM gerencia a aplicação desde a **concepção e governança** até a **manutenção e descontinuação (decommissioning)**. Ele integra o gerenciamento de requisitos, arquitetura, desenvolvimento, testes, rastreabilidade e gerenciamento de mudanças.

Arquitetura e Técnicas de Projeto de Software

A **Arquitetura de Software** é o conjunto de decisões críticas sobre a organização de um sistema. Ela envolve a escolha dos elementos estruturais, suas interfaces e o comportamento colaborativo entre eles. Enquanto o *Design* (projeto) se preocupa com detalhes de implementação de componentes individuais, a *Arquitetura* foca no "blueprint" global e nas restrições que guiarão todo o desenvolvimento.

Padrões Arquiteturais e Evolução



A escolha do padrão arquitetural impacta diretamente a capacidade de deploy e a performance do sistema. Historicamente, o modelo **Monolítico** (onde toda a aplicação reside em uma única unidade de execução) era o padrão pela sua simplicidade inicial de desenvolvimento e teste. No entanto, com a necessidade de escalabilidade horizontal e ciclos de entrega ultra-rápidos, surgiram os **Microserviços**. Neste modelo, a aplicação é decomposta em serviços pequenos, independentes e que se comunicam via rede (geralmente REST ou mensageria), permitindo que diferentes partes do sistema sejam atualizadas sem afetar o todo.

- **Arquitetura em Camadas (Layered):** Organiza o código em níveis de responsabilidade (Apresentação, Negócio, Persistência), facilitando a separação de interesses.
- **Arquitetura Hexagonal (Ports and Adapters):** Isola a lógica de negócio central de influências externas (bancos de dados, APIs de terceiros), permitindo que o núcleo do software seja testado de forma isolada e independente de tecnologias de infraestrutura.
- **Serverless e Event-Driven:** Focam na execução de funções disparadas por eventos específicos, otimizando o custo computacional e a resposta a estímulos em tempo real.

CONHECIMENTOS ESPECÍFICOS - BLOCO II

QUALIDADE DE SOFTWARE: GARANTIA DA QUALIDADE DE SOFTWARE; GERÊNCIA DE CONFIGURAÇÃO DE SOFTWARE (GIT); ESTES DE SOFTWARE (UNITÁRIO, INTEGRAÇÃO, FUNCIONAL, ACEITAÇÃO, DESEMPENHO, CARGA, VULNERABILIDADE); TÉCNICAS PARA APLICAÇÃO DE TESTES DE SOFTWARE (CAIXA-BRANCA, CAIXA-PRETA, REGRESSÃO E NÃO FUNCIONAIS); FERRAMENTAS PARA AUTOMATIZAÇÃO DE TESTES; TÉCNICAS DE REFATORAÇÃO DE SOFTWA; TRATAMENTO DO DÉBITO TÉCNICO; MÉTRICAS DE QUALIDADE DE CÓDIGO; CODE SMELL; AUDITORIA DE SISTEMAS

A qualidade de software é um dos principais fatores que determinam o sucesso de um sistema. Um software de qualidade não é apenas aquele que "funciona", mas aquele que funciona corretamente, é seguro, eficiente, fácil de manter e atende às necessidades do usuário.

No desenvolvimento moderno, garantir qualidade não é uma etapa final, mas um processo contínuo que envolve planejamento, testes, revisão de código, organização de versões e boas práticas técnicas. Este capítulo apresenta os principais conceitos relacionados à qualidade de software, de forma clara e introdutória, permitindo que o aluno compreenda os fundamentos que sustentam sistemas confiáveis e bem estruturados.

Garantia da Qualidade de Software

A Garantia da Qualidade de Software (Quality Assurance – QA) é o conjunto de atividades planejadas para assegurar que o software atenda aos padrões estabelecidos.

Ela não se limita à fase de testes. Envolve todo o ciclo de desenvolvimento, desde o levantamento de requisitos até a manutenção.

Entre os principais objetivos da garantia da qualidade, destacam-se:

Prevenir defeitos antes que eles ocorram.

Garantir conformidade com padrões e requisitos.

Melhorar continuamente os processos de desenvolvimento.

Reduzir riscos e retrabalho.

A qualidade é construída ao longo do processo, e não apenas verificada no final.

Gerência de Configuração de Software (GIT)

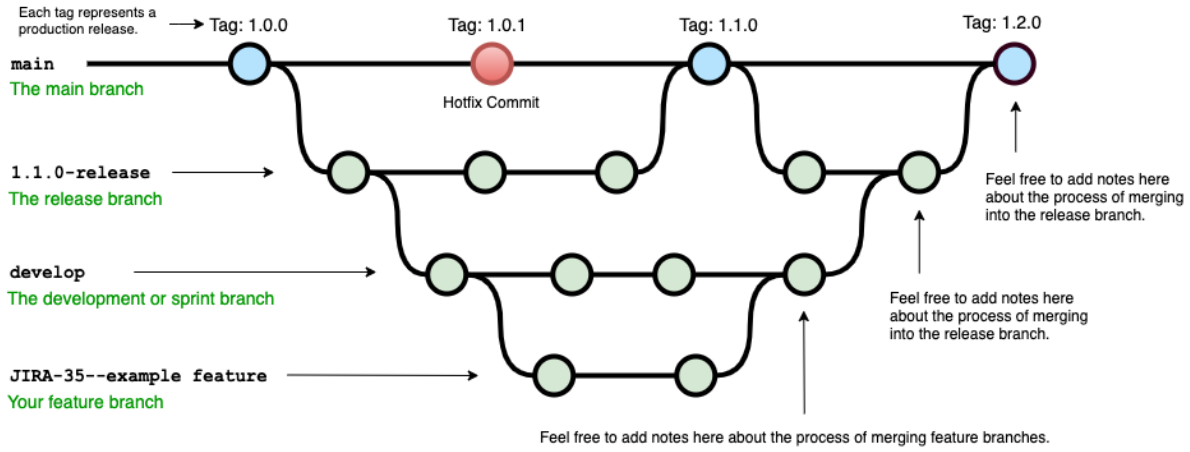
A Gerência de Configuração de Software é responsável por controlar e organizar as mudanças realizadas em um sistema ao longo do tempo.

Uma das ferramentas mais utilizadas para esse controle é o Git, sistema de versionamento que permite registrar alterações no código, manter histórico de versões e trabalhar em equipe de forma organizada.

Example Git Branching Diagrams

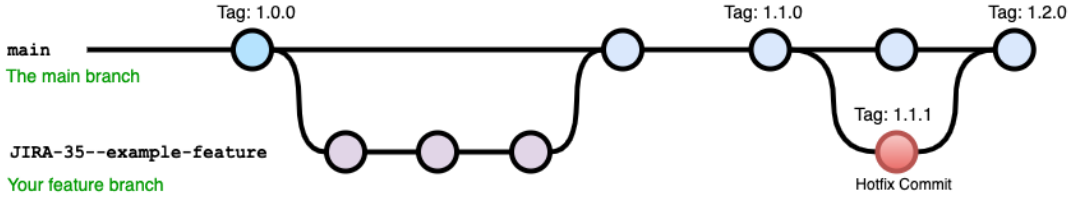
Example diagram for a workflow similar to "Git-flow" :

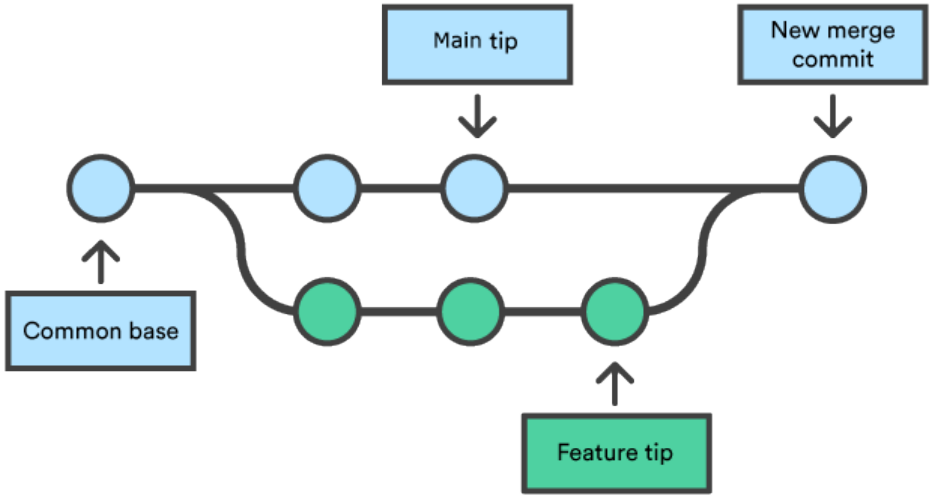
See: https://nvie.com/posts/a-successful-git-branching-model/



Example diagram for a workflow with a simpler branching model:

See: https://gist.github.com/jbenet/ee6c9ac48068889b0912 or https://www.endoflineblog.com/oneflow-a-git-branching-model-and-workflow





CONHECIMENTOS ESPECÍFICOS - BLOCO III

GERENCIAMENTO DE PRODUTOS DE SOFTWARE: GERENCIAMENTO DE PRODUTOS COM MÉTODOS ÁGEIS: SCRUM E KANBAN; MODELOS E TÉCNICAS DE GESTÃO DE PORTFÓLIO (SAFE): CARACTERÍSTICAS, OBJETIVOS, APLICABILIDADE E BENEFÍCIOS

AGILIDADE NO DNA DO PRODUTO: SCRUM E KANBAN

Historicamente, o software era desenvolvido no modelo "Cascata" (*Waterfall*), onde cada etapa (requisitos, design, código, testes) ocorria sequencialmente. O problema? Quando o software chegava ao cliente, meses depois, o mercado já havia mudado. Os métodos ágeis surgiram para quebrar esse ciclo, focando em entregas pequenas, frequentes e baseadas no feedback real dos usuários.

O Framework Scrum: Ritmo e Disciplina

O **Scrum** é um framework iterativo e incremental. Ele divide o trabalho em blocos de tempo fixos chamados **Sprints** (geralmente de 2 a 4 semanas). O coração do Scrum bate na previsibilidade e na melhoria contínua, estruturando-se em três pilares: transparência, inspeção e adaptação.

- **Papéis Fundamentais:** O **Product Owner (PO)** define *o que* será feito, priorizando o valor de negócio no *Product Backlog*. O **Scrum Master** atua como um facilitador, removendo impedimentos e garantindo que o time siga os valores ágeis. Os **Desenvolvedores** são o time multidisciplinar que transforma ideias em software funcional.
- **A Dinâmica das Cerimônias:** Tudo começa no *Sprint Planning*, onde o time decide o que cabe na próxima Sprint. Diariamente, a *Daily Scrum* alinha o progresso em 15 minutos. Ao final, a *Sprint Review* demonstra o que foi construído para os interessados, e a *Sprint Retrospective* analisa como o time pode trabalhar melhor no próximo ciclo.

O Método Kanban: Visualização e Fluxo Contínuo

Enquanto o Scrum é focado em "caixas de tempo" (Sprints), o **Kanban** é focado no **fluxo contínuo**. Originado no sistema de produção da Toyota, sua aplicação na TI foca em visualizar o trabalho e limitar a quantidade de tarefas em andamento para evitar gargalos.

- **O Quadro Kanban:** O trabalho é representado visualmente em colunas (ex: *To Do, Doing, Testing, Done*). Isso permite que qualquer pessoa identifique instantaneamente onde o fluxo está travado.
- **WIP (Work in Progress):** O segredo do Kanban é limitar o número de itens em cada coluna. Se o limite de "Testing" é 2 e já existem 2 tarefas lá, ninguém pode mover nada novo para essa coluna até que algo seja concluído. Isso força o time a focar em "terminar o que começou" em vez de "começar coisas novas", reduzindo o desperdício e aumentando a velocidade de entrega (*throughput*).

Scrum vs. Kanban: Qual escolher?

A escolha entre os dois depende da natureza do produto. O **Scrum** é excelente para produtos que precisam de um roteiro estruturado e onde o time pode se comprometer com metas de curto prazo. Já o **Kanban** brilha em ambientes de suporte, manutenção ou fluxos de trabalho muito dinâmicos, onde as prioridades mudam a cada hora e o objetivo é manter a "esteira" de produção sempre rodando sem interrupções artificiais.

Gestão de Portfólio e o Desafio da Escala (SAFe)

O gerenciamento de um único produto com um time de sete pessoas é um desafio tático; gerenciar um portfólio de dez produtos interdependentes com cinquenta times é um desafio estratégico. O **SAFe (Scaled Agile Framework)** surge como a resposta do mercado para organizações que precisam de governança, sincronização e previsibilidade sem perder a agilidade conquistada na base.

O que é o SAFe? Sincronizando o "Trem" Ágil

O SAFe é uma base de conhecimento de princípios e práticas comprovadas para aplicar Lean, Agile e DevOps em larga escala. Ele introduz o conceito de **Agile Release Train (ART)**, ou o "Trem de Liberação Ágil". Imagine que cada "vagão" desse trem é um time Scrum ou Kanban. Para que o produto final chegue à estação (o mercado) com sucesso, todos os vagões precisam rodar nos mesmos trilhos, na mesma velocidade e com a mesma direção.

- **Características e Objetivos:** O foco do SAFe não é apenas "fazer software rápido", mas sim o **Alinhamento** e a **Transparência**. Ele busca garantir que o que o desenvolvedor está codificando hoje esteja diretamente ligado aos objetivos estratégicos que o CEO definiu para o ano.
- **Cadência e Sincronização:** Diferente do Scrum individual, onde cada time pode ter sua própria duração de Sprint, no SAFe todos os times do "trem" operam na mesma cadência. Eles planejam juntos a cada 8 a 12 semanas em um evento massivo chamado **PI Planning (Program Increment Planning)**.

Níveis de Gestão e Aplicabilidade

O framework é modular, permitindo que a empresa escolha o nível de complexidade necessário:

- **Essential SAFe:** O nível básico, focado na coordenação de vários times que trabalham no mesmo produto (o ART).
- **Portfolio SAFe:** O nível mais alto, onde a diretoria decide onde investir o dinheiro da empresa. Aqui, a gestão não é sobre "tarefas", mas sobre **Épicos de Negócio** e fluxos de valor. O objetivo é garantir que o orçamento de TI esteja sendo gasto nas iniciativas que trarão o maior retorno sobre o investimento (ROI).

Benefícios e a Entrega de Valor

A grande vantagem do SAFe é a redução do "ruído" organizacional. Em empresas grandes e tradicionais, é comum que times trabalhem em funcionalidades que ninguém vai usar ou que entram em conflito com o trabalho de outro departamento.

- **Visibilidade:** O SAFe torna as dependências visíveis. Se o Time A precisa de uma API do Time B para entregar sua parte, isso é mapeado visualmente meses antes, evitando atrasos críticos na data de lançamento.
- **Time-to-Market:** Ao focar no fluxo de valor ponta a ponta (do conceito ao dinheiro), o framework ajuda a eliminar desperdícios burocráticos, permitindo que grandes corporações respondam a ataques de *startups* com agilidade competitiva.

Técnicas de Gestão de Portfólio e Métricas Ágeis

Ter uma lista de desejos (Backlog) infinita é fácil; o desafio real da TI é que a capacidade de desenvolvimento é sempre finita. A gestão de portfólio atua como o filtro estratégico que decide quais iniciativas receberão investimento, baseando-se no valor de entrega e no custo do atraso.

Priorização de Valor: A Técnica do WSJF

No modelo tradicional, quem gritava mais alto ou tinha o cargo mais alto (o famoso *HiPPO - Highest Paid Person's Opinion*) decidia a prioridade. No gerenciamento ágil de portfólio, utilizamos o **WSJF (Weighted Shortest Job First)**.

- **A Lógica do WSJF:** Esta técnica calcula o "Custo do Atraso" (*Cost of Delay*) dividido pelo "Tamanho do Trabalho".
- **O Objetivo:** Identificar tarefas que entregam um valor gigantesco em um tempo relativamente curto. Se uma funcionalidade pode trazer um milhão de reais em lucro e leva apenas duas semanas para ser feita, ela "fura a fila" de projetos de seis meses que trazem o mesmo retorno. O WSJF foca na economia do fluxo: o que nos dá o maior retorno com o menor esforço imediato?

Gestão de Dependências: O Mapa do Caminho

Em sistemas complexos (como um aplicativo de banco), o Time de Pagamentos não consegue entregar nada se o Time de Segurança não liberar o acesso aos dados. Gerenciar o portfólio é, em grande parte, gerenciar essas **interdependências**.

- **Program Board:** No SAFe e em outras metodologias de escala, utiliza-se um quadro visual onde fios (físicos ou digitais) conectam as tarefas de diferentes times. Se o Time A atrasar, o "fio" mostra imediatamente que o Time B e o Time C também serão impactados. Isso permite que o Gestor de Produto antecipe crises e renegocie prazos antes que o problema chegue ao cliente final.

Agilidade no DNA do Produto: Scrum e Kanban

Historicamente, o software era desenvolvido no modelo "Cascata" (Waterfall), onde cada etapa (requisitos, design, código, testes) ocorria sequencialmente. O problema? Quando o software chegava ao cliente, meses depois, o mercado já havia mudado. Os métodos ágeis surgiram para quebrar esse ciclo, focando em entregas pequenas, frequentes e baseadas no feedback real dos usuários.

O Framework Scrum: Ritmo e Disciplina

O Scrum é um framework iterativo e incremental. Ele divide o trabalho em blocos de tempo fixos chamados Sprints (geralmente de 2 a 4 semanas). O coração do Scrum bate na previsibilidade e na melhoria contínua, estruturando-se em três pilares: transparência, inspeção e adaptação.

Papéis Fundamentais: O Product Owner (PO) define o que será feito, priorizando o valor de negócio no Product Backlog. O Scrum Master atua como um facilitador, removendo impedimentos e garantindo que o time siga os valores ágeis. Os Desenvolvedores são o time multidisciplinar que transforma ideias em software funcional.

A Dinâmica das Cerimônias: Tudo começa no Sprint Planning, onde o time decide o que cabe na próxima Sprint. Diariamente, a Daily Scrum alinha o progresso em 15 minutos. Ao final, a Sprint Review demonstra o que foi construído para os interessados, e a Sprint Retrospective analisa como o time pode trabalhar melhor no próximo ciclo.

O Método Kanban: Visualização e Fluxo Contínuo

Enquanto o Scrum é focado em "caixas de tempo" (Sprints), o Kanban é focado no fluxo contínuo. Originado no sistema de produção da Toyota, sua aplicação na TI foca em visualizar o trabalho e limitar a quantidade de tarefas em andamento para evitar gargalos.

O Quadro Kanban: O trabalho é representado visualmente em colunas (ex: To Do, Doing, Testing, Done). Isso permite que qualquer pessoa identifique instantaneamente onde o fluxo está travado.

WIP (Work in Progress): O segredo do Kanban é limitar o número de itens em cada coluna. Se o limite de "Testing" é 2 e já existem 2 tarefas lá, ninguém pode mover nada novo para essa coluna até que algo seja concluído. Isso força o time a focar em "terminar o que começou" em vez de "começar coisas novas", reduzindo o desperdício e aumentando a velocidade de entrega (throughput).

Scrum vs. Kanban: Qual escolher?

A escolha entre os dois depende da natureza do produto. O Scrum é excelente para produtos que precisam de um roteiro estruturado e onde o time pode se comprometer com metas de curto prazo. Já o Kanban brilha em ambientes de suporte, manutenção ou fluxos de trabalho muito dinâmicos, onde as prioridades mudam a cada hora e o objetivo é manter a "esteira" de produção sempre rodando sem interrupções artificiais.

GOSTOU DESSE MATERIAL?

Imagine o impacto da versão **COMPLETA** na sua preparação. É o passo que faltava para garantir aprovação e conquistar sua estabilidade. Ative já seu **DESCONTO ESPECIAL!**

EU QUERO SER APROVADO!