

TIPOS DE METODOLOGIA DE PROJETO

Durante a formulação de um projeto, existem muitas informações e etapas que serão percorridas, as quais irão **variar de acordo com as finalidades** e metas idealizadas. Por isso, existem inúmeras metodologias de projeto, pois cada uma terá um foco de melhoria.

Mas, por consenso, costumamos agrupá-las em três categorias gerais: **Tradicional, Ágil e Híbrida**.

Tradicional

As metodologias tradicionais, também chamadas de **clássicas** ou **predictivas**, são as que priorizam o **controle e o planejamento** do projeto. Isto é, as etapas são mapeadas e estruturadas antes de iniciar a execução.

Fatores como prazos, custos, riscos e entregas são delimitados de modo detalhado, com atividades pensadas para meses de trabalho implementado. Então, geralmente, há um objetivo pontual a ser alcançado a longo prazo, necessitando de métodos específicos para garantir a fidelidade ao plano e que ocorra o mínimo de alterações possíveis.

As principais características dessa metodologia são:

- Escopo detalhado no início do projeto;
- Cronograma completo;
- Orçamento e custos estipulados;
- Processos mais seguros e conhecidos;
- Plano formal do projeto.

ÁGIL - (IMPORTANTE)

As metodologias ágeis, também chamadas de **adaptativas**, são aquelas que priorizam uma visão mais ampla do projeto, tendo um foco maior no valor da entrega do que no planejamento em si.

Inicialmente, elas eram direcionadas para a área de desenvolvimento de softwares.

Assim, a ideia é que o projeto seja estruturado ao longo de sua execução, adaptando-se as necessidades que surgirem com o tempo. Por isso, geralmente, os objetivos são definidos em **ciclos**, de 2 a 6 semanas, para que se possa ter entregas mais rápidas e pontuais.

Suas principais características são:

- Alta capacidade de adaptação;
- Visão geral do projeto;
- Escopo por ciclo;
- Alta produtividade (fator tempo e custo como determinante);
- Processos adaptáveis;
- Velocidade na tomada de decisões.

Metodologia Tradicional X Metodologia Ágil

MODELO TRADICIONAL

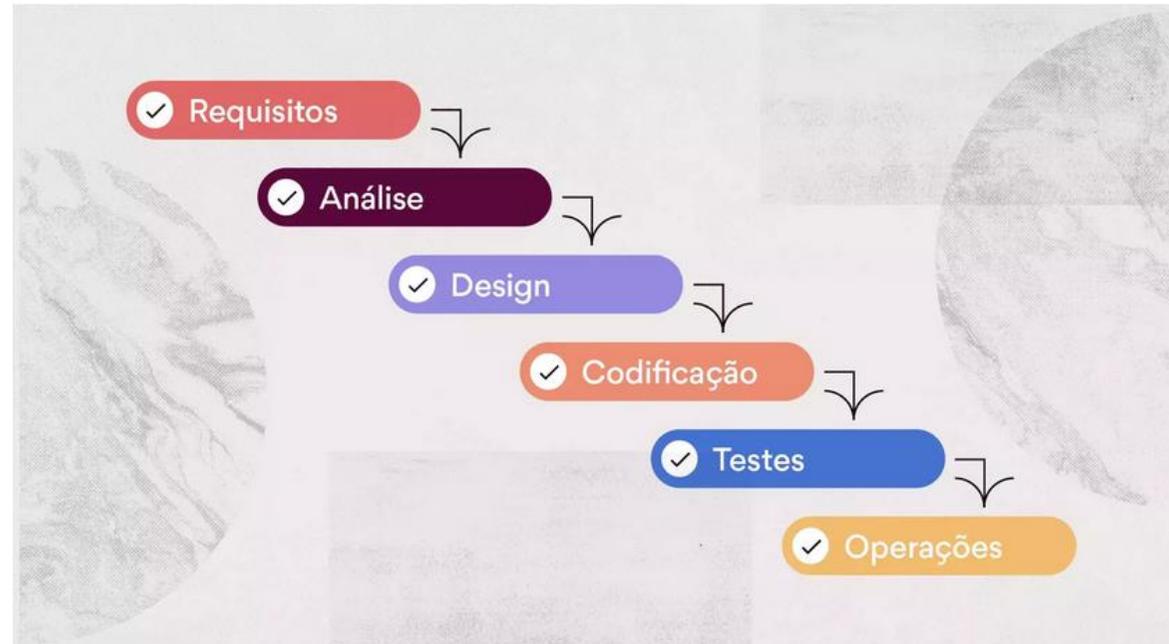


MODELO ÁGIL



1. Cascata

O que é: O modelo de cascata também é muito popular. Ao contrário do Agile, o método cascata é uma verdadeira metodologia, com alto grau de objetividade. A [metodologia de cascata](#), também conhecida como ciclo de desenvolvimento do software (Software Development Life Cycle, SDLC), é um processo linear no qual o processo flui de maneira similar a uma cascata, organizado em uma ordem sequencial.



Nesta abordagem, cada tarefa é conectada por uma dependência. Isto significa que cada tarefa deve ser concluída antes do início da próxima, assegurando a manutenção do trabalho no curso correto e a promoção de uma comunicação clara ao longo do processo.

Embora seja vista como uma abordagem tradicional por algumas organizações mais modernas, esta metodologia funciona bem ao criar um [planejamento de projeto](#) previsível e detalhado.

Quem deve usá-la: A metodologia cascata de gestão de projetos, por ser tão detalhada, é excelente para o trabalho em grandes projetos com grande número de participantes. A razão para tal é a presença de passos claros ao longo de todo o projeto e de dependências que permitem acompanhar o trabalho necessário para o alcance das metas.

2. Scrum

O que é: A metodologia [Scrum](#) envolve “Sprints” curtos, usados para criar um ciclo de projeto. Os ciclos duram de uma a duas semanas e são organizados com equipes de dez membros ou menos. Ela difere da metodologia de cascata, na qual as tarefas são divididas em dependências.

O Scrum é único por diversas razões, e uma delas é a figura do mestre do Scrum. Em outras palavras, ele é um gestor de projetos que lidera diariamente as reuniões, demonstrações, Sprints e retrospectivas de Sprints do Scrum, após a conclusão de cada Sprint. Estas reuniões visam unir os participantes do projeto e assegurar que as tarefas sejam concluídas dentro do prazo.

Embora o Scrum seja uma metodologia de gestão de projetos em si mesma, ele é frequentemente usado em associação com o modelo Agile, pois ambos compartilham princípios semelhantes, como a colaboração e a valorização dos indivíduos em detrimento dos processos.

Quem deve usá-la: As equipes que usam abordagens Agile devem usar, ou pelo menos experimentar a metodologia Scrum. Ao dividir os Sprints em equipes pequenas, esta abordagem pode funcionar bem tanto nas equipes pequenas quanto nas grandes.

3. Lean - (IMPORTANTE) -

O que é: a [metodologia Lean de gestão de projetos](#) tem o objetivo de reduzir desperdícios e criar uma estrutura simples para as necessidades do projeto. Em última análise, isto significa fazer mais com menos recursos, maximizando a eficiência e o trabalho em equipe.

Embora a redução de desperdícios originalmente se referisse a produtos físicos (desde o tempo em que Henry Ford usava este método e, posteriormente, a Toyota e a Motorola), hoje ele se refere às práticas dispendiosas. Ele é representado pelos três Ms:

- **Muda (desperdício):** práticas que consomem recursos mas não agregam valor.
- **Mura (inconstância):** ocorre quando há excesso de produção, com a geração de desperdício.
- **Muri (sobrecarga):** Ocorre quando os recursos estão sob elevado estresse.

Como gestor de projetos, o seu trabalho é prevenir a ocorrência dos três Ms, de maneira a melhorar a execução dos projetos e agilizar os processos. Ela é semelhante à abordagem de processo racional unificado (Rational Unified Process, RUP), a qual também visa à redução do desperdício. A diferença é que a RUP procura reduzir os custos de desenvolvimento, em vez das práticas dispendiosas.

Quem deve usá-la: Como o Lean tem como objetivo principal a redução do desperdício, é mais adequado para as equipes com dificuldades ou problemas de eficiência. Embora possa ter maior impacto em grandes organizações, ele pode ser útil para equipes de todos os tamanhos.

4. Kanban - (IMPORTANTE)

O que é: A metodologia [Kanban](#) representa os repositórios de projetos por meio de elementos visuais, em particular os quadros. Esta abordagem é usada por equipes Agile para [melhorar a visualização dos fluxos de trabalho](#) e do progresso do projeto, ao mesmo tempo em que reduz a probabilidade de ocorrência de gargalos. Ela também é frequentemente usada como uma ferramenta de software que permite a alteração e a movimentação dos quadros com fluidez entre os projetos, embora isto não seja obrigatório.

Muitas equipes usam este método de maneiras diferentes porque, ao contrário de outras metodologias, ele não apresenta um processo definido. O conceito a se ter em mente é que o Kanban se concentra nas principais tarefas do projeto, enquanto mantém o modelo geral simples.

Quem deve usá-la: Os quadros Kanban são excelentes para equipes de todos os tamanhos e, especificamente, para equipes predominantemente remotas. Isto se deve às qualidades visuais dos quadros Kanban, que auxiliam os membros da equipe a se manterem no rumo correto, onde quer que estejam.

Os quadros de Kanban podem e devem ser personalizados para a realidade dos projetos. Verifique abaixo 03 modelos de quadros de Kanban

Site recomendado: <https://kanbanflow.com>

A fazer	3 Análise		5 Desenvolvimento		4 Testes		Finalizado
Item de trabalho	Fazendo	Feito	Fazendo	Feito	Fazendo	Feito	
Item de trabalho	■	■	■	■	■	■	■
■		■	■	■	■		■
■			■	■	■		■
■				■	■		■
							■
							■
							■

EXEMPLO DE UM PROJETO

IDENTIFICAÇÃO

NOME DO PROJETO	Formulario TRR – Time de Resposta Rápida
DESCRIÇÃO	Criar uma maneira de cadastrar os acionamentos dos Médicos e o recebimento por parte deles, gerando indicadores e gráficos
ÓRGÃO SOLICITANTE	Coordenadoria - São Joaquim
NOME DO SOLICITANTE	Aristides
CONTATO DO SOLICITANTE	Ramal: 1534
DATA DE SOLICITAÇÃO	13/10/2021

OBJETIVO

Criar uma maneira de cadastrar os acionamentos dos Médicos e o recebimento por parte deles, gerando indicadores e gráficos para o Time de Resposta Rápida - TRR

JUSTIFICATIVA

Visando um fluxo de informações instantâneas

ESCOPO INICIAL DO PROJETO / LISTA DE FUNCIONALIDADES

ITEM	PRIORIDADE	ESCOPO SIMPLES DO PROJETO
1.	ALTA	CRIAR FORMULARIO PARA A PARTIR DE UM Nº DE ATENDIMENTO, REALIZAR O REGISTRO DE ACIONAMENTO DE MÉDICO COM DATA/HR DO ACIONAMENTO
2.	MEDIA	CRIAR FORMULARIO PARA RECEBIMENTO DE ACIONAMENTO PARA O MÉDICO COM DATA/HR DO RECEBIMENTO DO ACIONAMENTO
3.	MEDIA	GERAR INDICADORES A PARTIR DOS DADOS DE ACIONAMENTO E RECEBIMENTO
4.	BAIXA	GERAR GRÁFICOS DOS INDICADORES

FONTE DE CUSTEIO

Unidade São Joaquim e possivelmente outras unidades de internação da Santa Casa de Misericórdia de Sobral

ABRANGÊNCIA E IMPORTÂNCIA PARA A SANTA CASA

Colaboradores do Time de Resposta Rápida – TRR da Santa Casa de Sobral. Gestão da Santa Casa que receberá informações com mais facilidade e agilidade

DATA DE CONCLUSÃO ESPERADA (JUSTIFICAR)

Conclusão em no máximo noventa (90) dias após a aprovação do projeto por parte da diretoria.

MODELO DE ORGANIZAÇÃO DO ESCOPO DE TAREFAS DE UM PROJETO WEB

Nº	PRIORIDADE	TAREFA
1.	MÉDIA	DEFINIR DESIGN DO SITE
2.	MÉDIA	DEFINIR TECNOLOGIA DE BANCO DE DADOS
3.	MÉDIA	DEFINIR LINGUAGEM DE PROGRAMAÇÃO DO BACK END
4.	BAIXA	DEFINIR LINGUAGEM DE PROGRAMAÇÃO DO FRONT END
5.	ALTA	DEFINIR LÍDER DO PROJETO
6.	ALTA	DEFINIR LÍDER DA EQUIPE DE DESENVOLVIMENTO
7.	ALTA	DEFINIR LÍDER DA EQUIPE DE DESIGN
8.	ALTA	DEFINIR LÍDER DA EQUIPE DE TESTES
9.	ALTA	DEFINIR LÍDER DA EQUIPE DE BANCO DE DADOS
10.	ALTA	DEFINIR MEMBROS DA EQUIPE DE TESTES
11.	ALTA	DEFINIR MEMBROS DA EQUIPE DE DESIGN
12.	ALTA	DEFINIR MEMBROS DA EQUIPE DE TESTES
13.	MÉDIA	CRIAR LOGO DO SITE
14.	MÉDIA	DEFINIR NOME DO SITE/PROJETO
15.	MÉDIA	CRIAR A PÁGINA INICIAL E COLOCAR EM CONSTRUÇÃO
16.		
17.		
18.		
19.		
20.		
21.		
22.		

MODELO DE ORGANIZAÇÃO DE EQUIPES

Nº	NOME DO ALUNO	EQUIPE	FUNÇÃO
1.	FULANO	PROJETO	LÍDER DO PROJETO
2.	DELTRANO	DESIGN	LÍDER DE DESIGN
3.	NELTRANO	DESENVOLVIMENTO	LÍDER DE DESENVOLVIMENTO
4.			
5.			
6.			
7.			
8.			
9.			
10.			
11.			
12.			
13.			
14.			
15.			
16.			
17.			
18.			
19.			
20.			
21.			
22.			
23.			
24.			
25.			
26.			
27.			

(IMPORTANTE) - EQUIPES DE ENVOLVIDAS EM UM PROJETO DE DESENVOLVIMENTO DE WEBSITE

1. **EQUIPE DE GERENCIAMENTO DE PROJETOS:** Responsável pela coordenação do projeto, acompanhando e gerenciando a liderança das outras equipes e também é responsável pela comunicação com o cliente.
2. **EQUIPE DE DESIGN:** Responsável pela criação do design visual do site, incluindo a seleção de cores, fontes, layout e outros elementos visuais.
3. **EQUIPE DE DESENVOLVIMENTO FRONT-END:** Responsável por transformar o design em código HTML, CSS e JavaScript, tornando o site navegável e interativo.
4. **EQUIPE DE DESENVOLVIMENTO BACK-END:** Responsável por criar a lógica e a funcionalidade do site, incluindo integração de bancos de dados, autenticação, segurança e outros recursos avançados.
5. **EQUIPE DE TESTES:** Responsável por testar o site em diferentes dispositivos e navegadores, repassando os problemas para as outras equipes, garantindo assim que ele funcione corretamente e esteja livre de erros.

(IMPORTANTE) - TECNOLOGIAS ADOTADAS NOS PROJETOS DE SITES WEB

VISUAL STUDIO CODE: é um editor (quase uma IDE) de código-fonte gratuito e de código aberto, desenvolvido pela Microsoft. Ele possui uma ampla variedade de recursos e extensões para programação em diversas linguagens, além de suporte integrado ao controle de versionamento, depuração, terminal integrado e outras ferramentas de desenvolvimento. É uma das ferramentas mais populares entre desenvolvedores de software atualmente.

MySQL: é um sistema de gerenciamento de banco de dados relacional de código aberto, desenvolvido pela Oracle Corporation. Ele é amplamente utilizado em aplicativos da web, desde pequenos sites até grandes empresas, devido à sua eficiência, confiabilidade, escalabilidade e facilidade de uso. Ele suporta diversas linguagens de programação, como PHP, Java, Python, entre outras, e é compatível com diferentes sistemas operacionais. O MySQL é conhecido por sua alta performance em ambientes de alta carga de trabalho e por ter uma grande comunidade de usuários e desenvolvedores em todo o mundo.

HTML: é uma linguagem de marcação de texto utilizada para criar e estruturar conteúdo para a web. Ela permite que os desenvolvedores criem páginas web com texto, imagens, links, formulários e outros elementos interativos. O HTML é a base de todas as páginas da web e é interpretado pelos navegadores para renderizar o conteúdo em uma página da web

JavaScript: é uma linguagem de programação interpretada, utilizada para criar aplicativos dinâmicos e interativos para a web. É executada no lado do cliente, diretamente no navegador, permitindo a manipulação de elementos HTML e CSS em tempo de execução. Ele pode ser utilizado para validação de formulários, criação de animações, interações com APIs, entre outras funcionalidades.

CSS: é uma linguagem de estilo utilizada para definir a aparência e o layout dos elementos de uma página web. Ele permite que os desenvolvedores separem o conteúdo HTML da apresentação visual, permitindo uma maior flexibilidade e controle sobre o design de uma página. Com CSS, é possível definir a cor, fonte, tamanho, posição e outros atributos de estilo de um elemento HTML. O CSS é uma tecnologia de folhas de estilo que é interpretada pelos navegadores da web para renderizar a página em um layout estilizado e visualmente atraente

PHP: é uma linguagem de programação de código aberto amplamente utilizada para desenvolvimento de aplicativos da web dinâmicos. Ele é executado no lado do servidor e é capaz de se comunicar com bancos de dados, processar formulários e gerar conteúdo dinamicamente para as páginas da web. O PHP é compatível com diversos bancos de dados, como o MySQL, PostgreSQL e Oracle, e é uma das linguagens mais populares para desenvolvimento de sites e aplicativos da web.

AJAX: é uma técnica de desenvolvimento web onde precisar recarregar a página inteira. Ele permite que as interações do usuário com a página web sejam processadas de forma assíncrona, sem a necessidade de uma nova solicitação ao servidor. Com AJAX, é possível atualizar partes específicas de uma página, enviar e receber dados do servidor em segundo plano e melhorar a experiência do usuário em geral. Ele é frequentemente utilizado em conjunto com tecnologias como JavaScript, XML, JSON, HTML e CSS.

APACHE: O Apache é um servidor web de código aberto, amplamente utilizado em todo o mundo para hospedar e servir páginas da web e outros conteúdos na internet. Ele é um software gratuito, multiplataforma e altamente configurável que suporta diversos recursos avançados. O Apache é um dos servidores web mais populares e é frequentemente utilizado em conjunto com outras tecnologias, como PHP e MySQL.

XAMPP: é um pacote de software gratuito e de código aberto que contém os componentes necessários para criar e executar aplicativos da web localmente. Ele inclui o servidor Apache, banco de dados MySQL e interpretador para a linguagem de programação PHP. O XAMPP é frequentemente usado para desenvolvimento e teste de aplicativos da web em um ambiente local antes de serem implantados em um servidor de produção.

(IMPORTANTE) - DEFINIÇÕES IMPORTANTES

FRONT-END: é a parte de um aplicativo ou site da web que os usuários podem interagir diretamente. É a camada visual que os usuários veem e com a qual interagem, composta principalmente de HTML, CSS e JavaScript. O front-end é responsável por criar a interface do usuário, incluindo o design, o layout, a navegação e a interatividade.

BACK-END: é a parte do aplicativo ou site da web que os usuários não veem diretamente, mas que é essencial para o seu funcionamento. Ele consiste em um conjunto de tecnologias, linguagens de programação e frameworks que são usados para gerenciar o servidor, o banco de dados e a lógica de negócios do aplicativo/site. O back-end é responsável pelo processamento dos dados enviados pelo usuário, pela comunicação com o banco de dados e pela entrega dos resultados ao front-end. Desenvolvedores de back-end trabalham com tecnologias como PHP, Python, Ruby, Node.js etc.

(IMPORTANTE) - MOTIVOS PARA TRABALHAR EM EQUIPE

1. **Aprendizado:** Ao trabalhar em equipe, você tem a oportunidade de aprender com seus colegas, compartilhando conhecimento e experiência.
2. **Melhor solução:** Trabalhar em equipe ajuda a gerar ideias e perspectivas diferentes, resultando em soluções mais criativas e eficazes.
3. **Divisão de tarefas:** Trabalhar em equipe permite que as tarefas sejam divididas entre os membros, tornando o trabalho mais eficiente e permitindo que cada um se concentre em suas habilidades específicas.
4. **Aprendizado de habilidades sociais:** Trabalhar em equipe pode ajudar a desenvolver habilidades sociais, como liderança, comunicação, empatia e resolução de conflitos.
5. **Crescimento profissional:** Trabalhar em equipe pode ajudar a desenvolver habilidades profissionais que são altamente valorizadas pelos empregadores, como colaboração, trabalho em equipe e liderança.

(IMPORTANTE) - ORGANIZAÇÃO

A organização é um aspecto crítico do sucesso em projetos de desenvolvimento de software, garantindo que o projeto seja bem planejado, gerenciado de forma eficiente e desenvolvido de acordo com os padrões de qualidade estabelecidos. Veja abaixo alguns benefícios da organização nos projetos:

1. **Planejamento:** A organização ajuda a garantir que o projeto de software tenha um plano claro e bem definido, com objetivos, prazos e metas estabelecidos de antemão.
2. **Gerenciamento de recursos:** A organização permite que os recursos, como tempo, dinheiro e equipe, sejam gerenciados de forma eficiente, garantindo que o projeto seja concluído dentro do orçamento e prazo estabelecidos.
3. **Priorização de tarefas:** A organização ajuda a identificar as tarefas mais importantes e a priorizá-las de acordo, o que significa que o trabalho mais crítico pode ser concluído primeiro.
4. **Comunicação:** A organização facilita a comunicação entre os membros da equipe de desenvolvimento de software, permitindo que eles compartilhem informações importantes e coordenem o trabalho.

(IMPORTANTE) - COMUNICAÇÃO

Muito importante, pois permite que a equipe de desenvolvimento trabalhe de forma eficiente e colabore de maneira eficaz. Veja abaixo alguns motivos para se adotar uma boa comunicação nos projetos.

1. **Alinhamento de objetivos:** A comunicação eficaz permite que todos os membros da equipe entendam claramente os objetivos e metas do projeto, o que ajuda a garantir que todos estejam trabalhando em direção a um objetivo comum.
2. **Coordenação de atividades:** A comunicação ajuda a coordenar as atividades da equipe de desenvolvimento, permitindo que os membros trabalhem juntos de forma harmoniosa e eficiente.
3. **Solução de problemas:** A comunicação é essencial para resolver problemas e desafios que surgem ao longo do processo de desenvolvimento de software. Isso ajuda a garantir que o projeto continue avançando de maneira suave e eficaz.
4. **Tomada de decisões:** A comunicação eficaz é essencial para tomar decisões informadas durante o processo de desenvolvimento de software. Isso ajuda a garantir que a equipe esteja alinhada em relação às decisões tomadas e ajuda a minimizar a possibilidade de erros.
5. **Feedback:** A comunicação permite que os membros da equipe de desenvolvimento forneçam feedback uns aos outros, o que é importante para melhorar a qualidade do software em desenvolvimento.