

# CONCEITOS BÁSICOS RELACIONADOS COM O DESENVOLVIMENTO DE SOFTWARE

## SOFTWARE

Seria possível a sua vida hoje sem o smartphone?

Acredito que a maioria em uma pesquisa iria responder um enfático “não”!

O que você acha que torna o telefone móvel tão atrativo? O software?

É provável, pois, não descartando a importância do hardware, o software permite uma relação de usabilidade interativa e intuitiva, gerando um uso intensivo. Essa competência é resultado de trabalho multidisciplinar árduo de engenheiros de softwares, administradores de banco de dados, web designers, entre outros profissionais.

Neste contexto, vamos iniciar nosso estudo sobre a extraordinária área de conhecimento Engenharia de Software, que tem como principal produto o **software**.

### Atenção

Destacamos que a bibliografia Pressman (2016) é uma referência mundial nessa área.

O conceito de software a seguir define esse importante produto.

**Software consiste em:**

**(1) instruções (programa de computador) que, quando executadas, fornecem características, funções e desempenho desejados; (2) estruturas de dados que possibilitam aos programas**

**manipular informações adequadamente; e (3) informação descritiva, tanto na forma impressa quanto na virtual, descrevendo a operação e o uso dos programas.**

(PRESSMAN, 2016).

Implicitamente, o software relaciona-se com o hardware. Essa relação gerou a denominada “Crise do Software”, sintetizada pela afirmativa a seguir de Edsger.

**A maior causa da Crise do Software é que as máquinas se tornaram várias ordens de magnitude mais potentes! Em termos diretos, enquanto não havia máquinas, programar não era um problema; quando tivemos computadores fracos, isso se tornou um problema pequeno; e agora que temos computadores gigantescos, programar tornou-se um problema gigantesco.**

(DIJKSTRA, 1972).

É importante destacar que o desenvolvimento tecnológico do hardware nos últimos anos permitiu o desenvolvimento de softwares cada vez mais complexos, tendo um forte impacto na indústria de software. Como exemplo, podemos apresentar a substituição do paradigma estruturado pelo paradigma orientado a objetos, baseado na programação orientada a objetos que permite o reuso intensivo de especificação, bem como uma melhor manutenibilidade e, como consequência, o desenvolvimento de softwares mais complexos.

Vamos enfatizar uma das principais finalidades do software: a geração de informação. A figura 1 ilustra de forma implícita os principais componentes tecnológicos de um Sistema de

Informação, i.e., hardware, software, sistema gerenciador de banco de dados, redes de comunicação e serviços, sendo os referidos componentes interdependentes. O ambiente empresarial está representado pela respectiva pirâmide funcional, cabendo ao componente software, o importante papel de agregar valor aos dados quando da geração de informações aos diferentes níveis de gestão – operacional, gerencial e estratégico, pois essa pode ser utilizada em um processo de tomada de decisão ou no controle de funções empresariais, tais como financeiro, recursos humanos e outras.

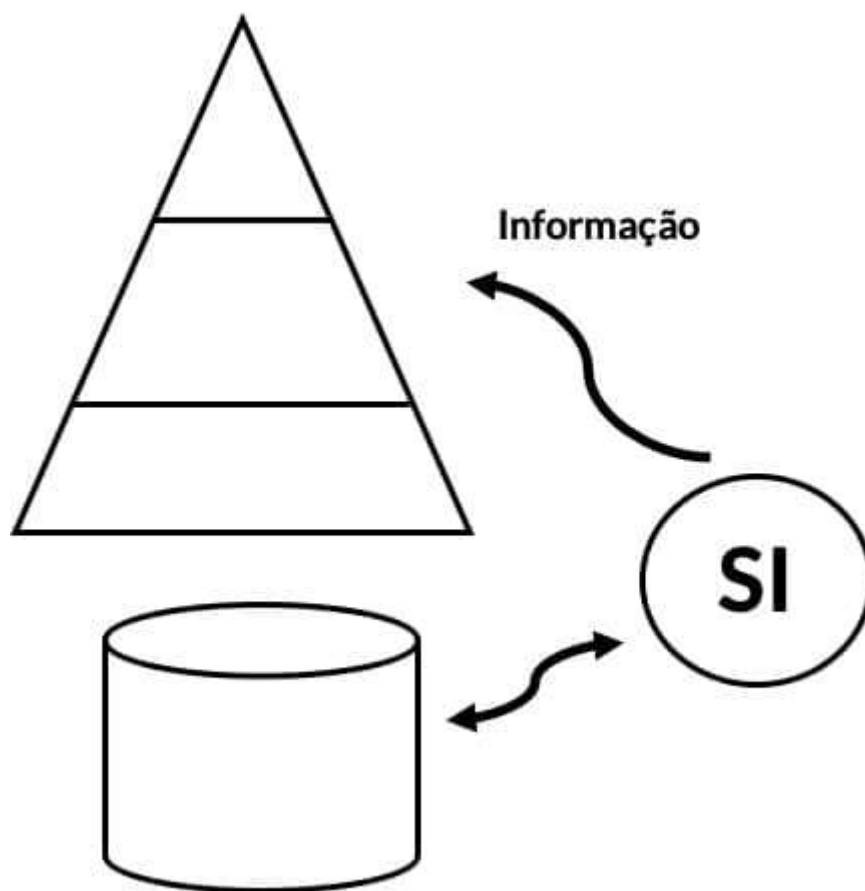


Figura 1 – Sistema de Informação x Software.

Um segundo exemplo, vamos considerar a complexidade do desenvolvimento de um software embarcado em uma aeronave com 500 pessoas e que realiza o seu controle total, em que um defeito pode ter um impacto altamente negativo.

Vamos imaginar o que seria do piloto sem as informações do voo no painel de controle.

Em ambos exemplos, observamos que o software apresenta dois papéis distintos, o primeiro como um produto a ser utilizado pelos usuários, o segundo como veículo que distribui o produto, pois a comunicação entre os diversos componentes de um sistema de informação ocorre por meio de sistemas operacionais, software de comunicação entre outros. O software distribui o produto mais importante da nossa era – a **Informação** (PRESSMAN, 2016).

Você consegue imaginar quais os desafios atuais de um engenheiro de software?

Destacamos que os desafios incluem sete grandes categorias:

## **SOFTWARE DE SISTEMA**

Camadas de software que atendem a outros softwares, tais como, sistemas operacionais, drivers e outros.

## **SOFTWARE DE APLICAÇÃO**

Inclui software com escopo específico, tais como, sistemas de gestão empresarial (ERP).

## **SOFTWARE DE ENGENHARIA/CIENTÍFICO**

Inclui software aplicado às áreas de engenharia e científica, tal como, software para cálculo estrutural na área de engenharia civil ou processamento de imagem.

## **SOFTWARE EMBARCADO**

Instalado em produtos com funções específicas, tal como, o controle de um veículo com informações disponíveis no painel digital.

## **SOFTWARE PARA LINHA DE PRODUTOS**

Projetado com determinado conjunto de funcionalidades e utilizado por diferentes clientes, por exemplo, sistema emissor de nota fiscal.

## **APLICAÇÕES WEB/APLICATIVOS MÓVEIS**

Software específico para dispositivo móvel.

## **SOFTWARE DE INTELIGÊNCIA ARTIFICIAL**

Utilizam técnicas de inteligência artificial, tais como, sistema especialistas, redes neurais, aprendizado de máquinas e outros.

# **ENGENHARIA DE SOFTWARE**

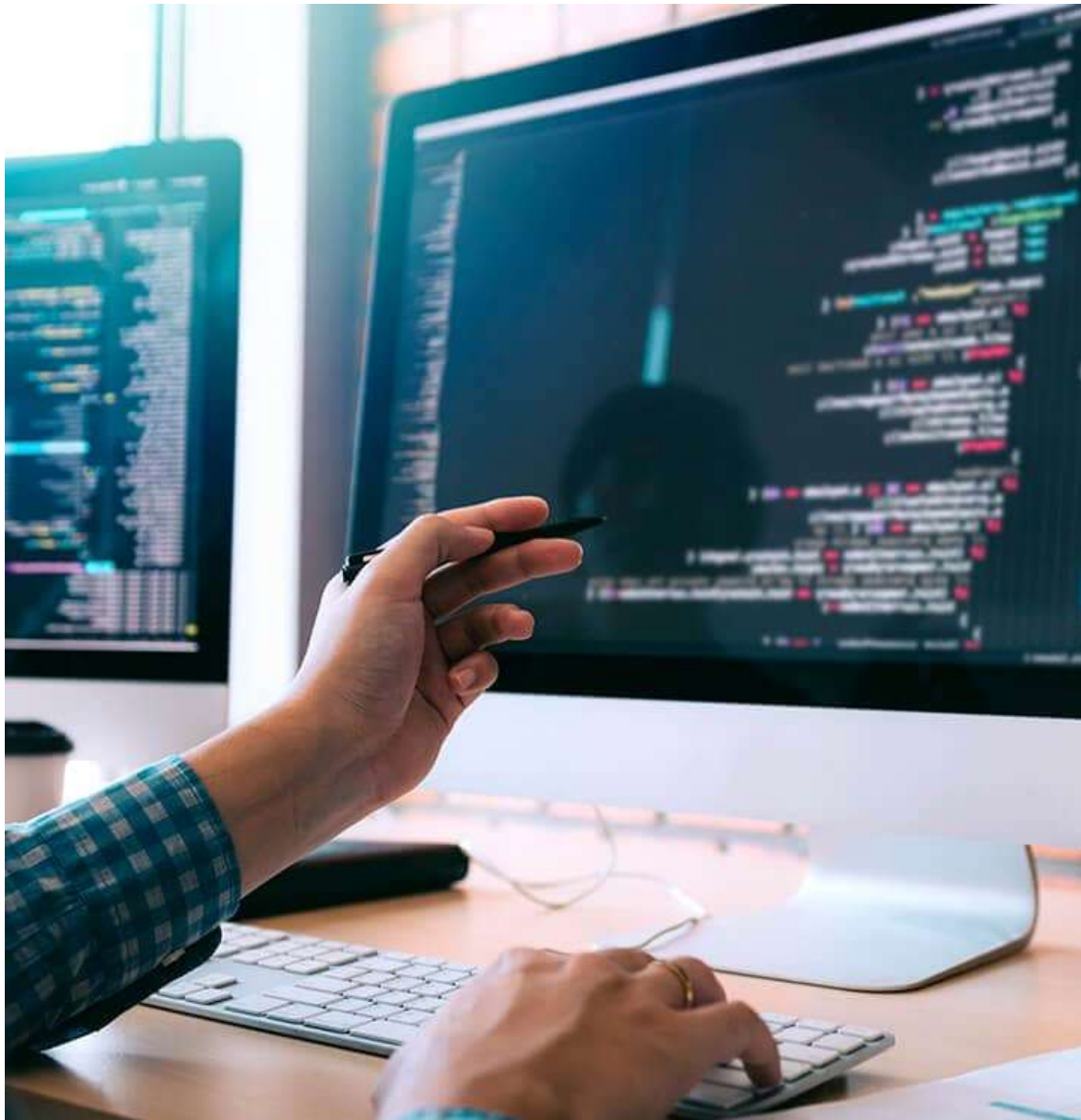
Vimos nos exemplos do software embarcado em uma aeronave ou controlando o tráfego aéreo uma característica comum: a complexidade. A melhor tratativa para a **complexidade** é a aplicação de metodologia que permita a decomposição do problema em problemas menores de forma sistemática, cabendo à engenharia essa sistematização.

Uma premissa básica é que a engenharia permite a solução de problemas e, quanto mais complexo um produto a ser gerado, mais a engenharia faz-se necessária.



Vamos ver no caso da construção de uma simples casa, talvez um engenheiro resolva o problema; e no caso de um edifício inteligente com vários andares? Neste caso, o problema tornou-se multidisciplinar, ou seja, serão necessários vários profissionais de várias áreas, e.g., arquiteto, engenheiros civis, mecânicos, elétricos, de software etc. A construção do prédio torna-se inviável caso não haja uma tratativa sistemática por meio da aplicação das melhores práticas da engenharia.

A mesma correlação pode ser aplicada quando o produto a ser gerado é o software. No caso do software, aplica-se a Engenharia de Software.



O desenvolvimento de software deve submeter-se aos mesmos princípios aplicados nas engenharias tradicionais, sendo um produto distinto em função da sua intangibilidade que o associa, muitas vezes, somente aos códigos dos programas de computador. Entretanto, diferentemente de outros produtos de outras engenharias em produção, possui alta volatilidade em função de constantes evoluções na tecnologia e nos requisitos, agregando ao software uma complexidade adicional.

A primeira referência ao termo “Engenharia de Software” foi em 1968, em uma conferência sobre o tema, sob responsabilidade do Comitê de Ciência da NATO (North Atlantic Treaty Organization). Nesse contexto, o Guide to the Software

Engineering Body of Knowledge Version 3.0 (SWEBOK Guide V3.0) apresenta a seguinte definição para a Engenharia de Software:

**A aplicação de uma abordagem sistemática, disciplinada e quantificável para o desenvolvimento, operação e manutenção de software, que é a aplicação de engenharia ao software.**

## **TECNOLOGIA EM CAMADAS**

A Engenharia de Software é uma tecnologia em camadas, como ilustra a figura 2. Vejamos as descrições das referidas camadas:

### **CAMADA DE QUALIDADE**

Garante que os requisitos que atendem às expectativas do usuário serão cumpridos.

### **CAMADA DE PROCESSO**

Determina as etapas de desenvolvimento do software.

### **CAMADA DE MÉTODOS**

Define, por exemplo, quais as técnicas de elicitação de requisitos, os artefatos gerados em função da técnica de modelagem adotada, tal como, modelo de casos de uso ou de classes.

### **CAMADA DE FERRAMENTAS**

Estimula a utilização de ferramentas “CASE” (Computer-Aided Software Engineering) no desenho dos diversos artefatos ou mesmo na geração automática de código, entre outras aplicações; a tecnologia CASE está disponível para uso em todas as etapas do processo de desenvolvimento de software.





Figura 2 - Camadas da Engenharia de Software.

Considerando um produto tangível para todos nós, você poderia imaginar que um dia construirá sua casa própria? Vamos imaginar que sim. O que você precisaria para levar em frente esta construção? Um processo?

Certamente, pois terá que organizar as etapas da construção: fundação, estrutura, cobertura, alvenaria, instalações etc. Agora a construção da casa fica mais fácil.

Da mesma maneira, a construção de um software necessita de processo; esse deverá ocorrer de acordo com as melhores práticas estabelecidas pela Engenharia de Software.

É importante destacar que a base da Engenharia de Software é a **camada de processo**.

## PROCESSO DE SOFTWARE

Processo é uma sequência de etapas que permitem a geração de um produto, no nosso caso, o software. O processo permite uma melhor tratativa em relação à complexidade de obtenção de um determinado produto, pois, na maioria das vezes, é um trabalho multidisciplinar realizado por analistas, programadores, gerentes de projeto, gerentes de teste e outros profissionais.

Assim como em todas as engenharias, a Engenharia de Software possui uma diversidade de modelos para processos de desenvolvimento de software que atendem a diferentes problemas.



Um requisito importante na seleção de um processo de software é a complexidade. Basicamente, quanto maior a complexidade de um sistema, mais formal deve ser o processo adotado. Importante: a qualidade é a camada base que sustenta a camada processo.

Uma metodologia de processo genérica compreende cinco atividades (figura 3).



Figura 3 - Metodologia do Processo.

## COMUNICAÇÃO

As primeiras atividades de um processo de software requerem uma comunicação intensiva com os usuários, buscando o entendimento do problema, a definição de objetivos para o projeto, bem como a identificação de requisitos.

## PLANEJAMENTO

Destaca-se nesta atividade a área de conhecimento Gerenciamento de Projeto, que permitirá a elaboração de um Plano de Gerenciamento do Projeto de forma sistemática, tendo

como entrega importante o cronograma que inclui as atividades a serem desenvolvidas no referido projeto, contemplando diferentes áreas de conhecimento.

O processo de desenvolvimento de software disponibiliza as principais atividades que irão compor o Plano de Gerenciamento do Projeto, sendo esse plano executado e monitorado.

## **MODELAGEM**

A engenharia tem como melhor prática a geração de modelos, tal como a planta baixa de uma casa. A maioria desses modelos gráficos, denominados de diagramas na Engenharia de Software, podem ser complementados por descrições textuais. Como exemplo, apresentamos o modelo de casos de uso que inclui diagramas de casos de uso, artefatos gráficos e descrições de cenários dos casos de uso, artefatos textuais.

## **CONSTRUÇÃO**

A partir dos modelos gerados, é realizada a construção ou implementação do software, portanto, os modelos determinam o comportamento do software. Essa atividade inclui a codificação e os testes de software de acordo com o planejado.

## **ENTREGA**

Ao final, ocorre o objetivo de um plano de projeto de software, i.e., a entrega do produto em produção de acordo com o planejado.

As atividades básicas apresentadas são complementadas pelas denominadas “atividades de apoio”, que incluem:

### **Controle e acompanhamento de projeto**

Dentro do princípio de que “não se controla o que não se mede”, a etapa do Gerenciamento de Projeto Monitoramento e Controle

permite verificar se a execução está de acordo com o planejado, pois, caso seja identificada alguma não conformidade, ações corretivas devem ser implementadas.

## **Administração de riscos**

Ênfase à área de Gerenciamento de Risco, de modo que qualquer evento, positivo ou negativo que possa impactar o desenvolvimento do projeto, seja tratado.

## **Garantia da qualidade de software**

Ênfase à área de Gerenciamento da Qualidade, a fim de garantir que os requisitos do projeto sejam atendidos.

## **Revisões técnicas**

Atividade intrínseca da qualidade, pois todos os artefatos devem ser testados, inclusive, processos, modelos, código do software e outros.

## **Medição**

Outra atividade da qualidade que permite a definição de métricas para avaliar as várias atividades durante o desenvolvimento do software.

## **Gerenciamento da configuração de software**

Inclui os efeitos das mudanças, por exemplo o gerenciamento de versionamento do software em um processo iterativo e incremental, onde um defeito em produção pode surgir em uma versão anterior à desenvolvida no momento, sendo a manutenção realizada na versão anterior e propagada para a versão atual. Este procedimento, muitas vezes, exige uma ferramenta que automatize a solução deste problema, que pode ser bem complexo.

## **Gerenciamento da capacidade de reutilização**

O reuso de software deve ser um objetivo persistido por quem desenvolve o software. No paradigma estruturado, existiam as bibliotecas de funções que permitiam a reutilização de código em várias aplicações com possibilidades bem limitadas e com o advento do paradigma orientado a objetos, o reuso ficou mais sofisticado em função de mecanismos, e.g. herança, que possibilitam uma melhor componentização do software.

## **Preparo e produção de artefatos de software**

Um processo de software encadeia uma série de atividades, sendo que estas atividades possuem métodos próprios para a geração de artefatos que necessitam ser documentados, e.g., modelo de casos de uso.

Cabe ao engenheiro de software documentar cada atividade que será aplicada no processo de desenvolvimento de software em função da natureza do problema (complexidade), das características das pessoas que realizarão o trabalho e dos usuários envolvidos.

## **RESUMINDO**

Neste módulo, podemos destacar a importância do software atualmente, bem como da complexidade no seu desenvolvimento. A aplicação da Engenharia de Software permite lidar com a referida complexidade, pois melhores práticas podem ser aplicadas de forma a gerar um produto “software” que atenda às necessidades para as quais foi projetado.

Destacamos que a Engenharia de Software é uma tecnologia em camadas, ou seja, com foco na qualidade, processo, métodos e ferramentas. Cabe enfatizar que a base da Engenharia de Software é a camada de processo, por isso foram descritas as principais atividades genéricas que devem compor um processo de software: comunicação, planejamento, modelagem, construção e entrega.