# Tipos de Arquitetura

4 tipos:

* Software
* Solução
* Tecnológica
* Corporativa

## Arquitetura Tecnologia

* Especialidade em tecnologias específicas de mercado
* Geração de valor baseado em especialidades
* Diversidade de processionais especialistas
  + Arquiteto Elastic
  + Arquiteto Java
  + DBA

## Arquitetura Corporativa

* Impacta estrategicamente a organização como um todo
* Avaliação de custos
* Avaliação de possíveis novas tecnologias
* Padronização de tecnologias
* Planejamento de grandes implantações
  + Sistemas “core”
  + Migrações

## Arquitetura de Solução

* Fica entre a área de negócios e software
* Transformar requisitos de negócio em soluções de software
* Desenhos arquiteturais da solução de um software para reproduzir como ele irá funcionar
  + C4
  + UML
  + BPMN
* Analisa os impactos comerciais em relação a uma escolha de tecnologia
* Pode participar do processo comercial de pré-venda e venda
* Analisa os impactos dos custos para o negócio

## Arquitetura de Software

* É uma “disciplina” da engenharia de software
* Diretamente ligada ao processo de desenvolvimento de software
* Afeta diretamente na estrutura organizacional da empresa
  + Formação dos times
  + Estrutura dos componentes do software
  + “Organizações que desenvolvem sistemas de software tendem a produzir sistemas que são cópias das estruturas de comunicação dessas empresas. (Melvin Conway)”

Arquitetura de Software, é a relação entre os objetivos de negócio e suas restrições com os componentes a serem criados e suas responsabilidades visando sua evolução do software.

“É a organização fundamental de um sistema e seus componentes, suas relações, seu ambiente, bem como os princípios que guiam seu design e evolução.” (IEEE Standard 1471)

O processo de arquitetar um software estabelece que o que está sendo desenvolvido faça parte de um conjunto maior (nesse caso o próprio negócio).

## Papel do Arquiteto de Software

* Transformar requisitos de negócio em padrões arquiteturais
* Orquestrar pessoas desenvolvedores e experts de domínio
* Entender de forma profunda conceitos e modelos arquiteturais
* Auxilia na tomada de decisão nos momentos de crise
* Reforça boas práticas de desenvolvimento
* Code reviews

Apesar de nem todas as organizações possuírem o cargo de arquiteto de software, normalmente profissionais mais experientes como desenvolvedores seniors e tech leads acabam realizando esse papel baseado em suas experiências anteriores.

Há empresas que apesar de não possuírem o cargo de arquiteto de software, possuem um departamento de arquitetura que auxilia os diversos times da organização com questões arquiteturais.

## Vantagens de aprender arquitetura de software

* Poder navegar da visão macro para visão micro de um ou mais softwares
* Entender quais são as diversas opções que temos para desenvolver a mesma coisa e escolher a solução para determinado contexto
* Pensar a longo prazo no projeto e sua sustentabilidade
* Tomar decisões de forma mais fria e calculada evitando assim ser influenciado por “hypes” de mercado
* Mergulho em padrões de projeto e de desenvolvimento e suas boas práticas
* Ter mais clareza do impacto que o software possui na organização como um todo
* Tomar decisões com mais confiança

## Arquitetura vs Design de Software

Arquitetura: Escopo global ou alto nível

Design: Escopo local

“Atividades relacionadas a arquitetura de software são sempre de design. Entretanto, nem todas as atividades de design são sobre arquitetura. O objetivo primário da arquitetura de software é garantir que os atributos de qualidade, restrições de alto nível e os objetivos de negócio, sejam atendidos pelo sistema. Qualquer decisão de design que não tenha relação com este objetivo não é arquitetural. Todas as decisões de desing para um componente que não sejam “visíveis” fora dele, geralmente, também não são”

## Sustentabilidade

* Desenvolver software é caro
* Software resolve uma “dor”
* Software precisa se pagar ao longo do tempo
* Acompanhar a evolução do negócio
* Quanto mais tempo o software fica no ar, mas retorno gera
* A solução precisa ser arquitetada

## Pilares da arquitetura de software

* Estruturação
  + Fácil evolução, componentização para atender os objetivos de negócio
* Componentização
* Relacionamento entre sistemas
* Governança (padronização, regras, documentação, boilerplate e etc)

## Requisitos arquiteturais (RAs)

* Performance
* Armazenamento de dados
* Escalabilidade
* Segurança
* Legalidade
* Auditoria
* Marketing

# Características Arquiteturais

* Operacionais
* Estruturais
* Cross-Cutting

## Características Operacionais

* Disponibilidade (o qual disponível quero deixar esse sistema? 24x7)
* Recuperação de desastres
* Performance (latência, capacidade de receber requisições...)
* Recuperação (backup, quando fui a última vez que foi testado?)
* Confiabilidade e segurança
* Robustes
* Escalabilidade (verticalmente é quando aumentamos o recurso computacional das nossas máquinas e horizontalmente quando aumentamos o número de máquinas). Procurar por 12 factors.