



# ATTRIBUTE DRIVEN DESIGN

Método sistemático de projeto de arquitetura

Grupo:

Arthur Nucada Félix de Souza - 202201683

José Alves de Oliveira Neto - 202201699

Lucas Gabriel Nunes Alves - 202201703

Sophia Fernandes Almeida - 202201713

Victor Martins Vieira - 202204532

# EXPLICANDO A ORIGEM

A origem do Attribute Driven Design (ADD) pode ser rastreada até a década de 90.

Método desenvolvido pelo Software Engineering Institute (SEI) do Carnegie Mellon University.

O (ADD) foi originalmente desenvolvido para aplicações de software de missão crítica, mas tem sido utilizado em outros tipos de projetos de software desde então.

Ele foi criado como uma abordagem para o projeto de software que se concentra em identificar e modelar atributos de um sistema que são importantes para seus usuários e stakeholders, com o objetivo de criar sistemas que sejam altamente adaptáveis, fáceis de manter e escaláveis, e que atendam às necessidades dos usuários e stakeholders.

1

PLAN

2

DO

3

CHECK

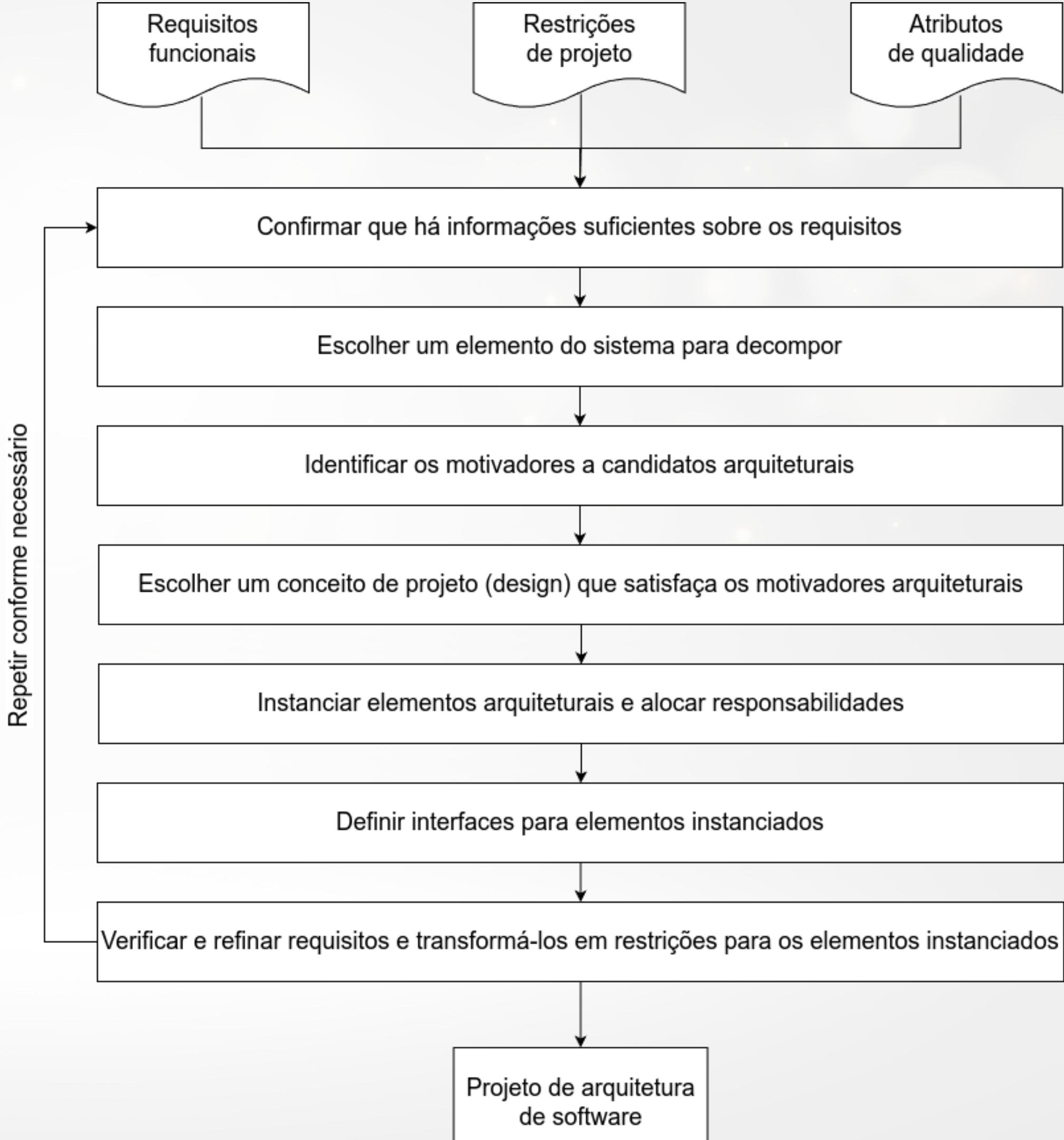
Os atributos de qualidade e as restrições de projeto são considerados para selecionar quais tipos de elementos serão usados na arquitetura

Elementos são implementados para satisfazer atributos de qualidade assim como os requisitos funcionais

A arquitetura resultante é analisada para determinar se os requisitos foram atendidos

# VISÃO GERAL O PROCESSO

- 8 passos;
- Processo recursivo.
- Requisitos e restrições como entrada;
- Projeto de arquitetura como saída.



# **ENTRADAS**

## **REQUISITOS FUNCIONAIS**

Funções que o sistema deve prover para atender as necessidades declaradas ou implícitas dos stakeholders quando utilizado sob condições específicas.

## **RESTRIÇÕES DE PROJETO**

Decisões sobre o projeto de um sistema que devem ser incorporadas ao projeto final do sistema. Geralmente, pré-determinam o resultado de um aspecto do projeto.

## **ATRIBUTOS DE QUALIDADE**

São requisitos que indicam graus de propriedades que um sistema deve exibir.



# PASSO 1 - INFORMAÇÕES SUFICIENTES

Nessa etapa, os seguintes passos são realizados:

- A importância de cada requisito deve ser classificada pelos *stakeholders*;
- A partir da lista com os requisitos classificados por prioridade, determinar quais elementos devem ser o foco do projeto;
- Considerar elementos e seus impactos sobre a arquitetura em ordem decrescente de importância para os *stakeholders*;
- Cada requisito de qualidade deve ser expresso sob a forma de estímulo e resposta.

Durante essa etapa, não são realizadas decisões de projeto (*design*).

# PASSO 2 - ESCOLHA DO ELEMENTO

- Na primeira iteração do processo, o elemento a ser decomposto é o sistema como um todo;
- A partir da segunda iteração, a escolha do elemento a ser decomposto é baseada no(s):
  - Atual conhecimento da arquitetura;
  - Risco e dificuldade;
  - Critério de negócio;
  - Critério Organizacional.
- Durante essa etapa, não são realizadas decisões de projeto (*design*).

# PASSO 3 - IDENTIFICAR MOTIVADORES

Nessa etapa, os seguintes atividades são realizadas:

- Classificar requisitos de acordo com seu impacto sobre a arquitetura;
- Agrupar prioridades baseado na sua importância para os stakeholders E no seu impacto sobre a arquitetura.
  - A classificação pode ser simples, como: alto, médio e baixo.
- Selecionar cinco ou seis requisitos com alta prioridade como os motivadores de candidatos arquiteturais.
- Durante essa etapa, não são realizadas decisões de projeto (*design*).



# PASSO 4 - ESCOLHER CONCEITO DE DESIGN

Nessa etapa, as seguintes atividades são realizadas:

- Identificar os principais interesses/benefícios de *design* dos motivadores a candidatos arquitetrais;
- Criar uma lista de padrões que atendem a cada interesse.
  - Identificar e estimar o valor que cada parâmetro do padrão traz à solução.
- Selecionar os padrões mais apropriados à satisfação dos motivadores;
  - Criar uma matriz com prós e contras para cada candidato e motivador;
  - Escolher os padrões que mais se aproximam de satisfazer os motivadores.
- Considerar como os padrões interagem entre si;
- Descrever os padrões selecionados a partir de diferentes visões arquiteturais;
- Encontrar e resolver inconsistências no conceito do projeto (*design*).

# PASSO 5 - INSTANCIAR EL. ARQUITETURAIS E ALOCAR RESPONSABILIDADES

Nessa etapa, os seguintes passos são realizados:

- Criar uma instância de cada tipo de elemento selecionado no passo anterior;
- Atribuir responsabilidades aos elementos-filhos de acordo com seu tipo;
- Alocar as responsabilidades do elemento pai entre os filhos de acordo com a lógica e as propriedades do elemento;
- Criar instâncias adicionais de tipos de elementos se houver diferenças entre os atributos de qualidade atribuídos ao elemento ou se quiser cumprir outros requisitos de qualidade;
- Revisar as decisões de projeto e confirmar que todas as decisões relevantes foram tomadas;
- Analisar e documentar as decisões de projeto usando as três perspectivas: módulo, componente-conector e Alocação.

# PASSO 5 - INSTANCIAR EL. ARQUITETURAIS E ALOCAR RESPONSABILIDADES

Nessa etapa, as seguintes decisões e considerações de *design* são atingidas:

- Quantos elementos de cada tipo serão instanciados e suas dependências;
- Elementos de infraestrutura para suportar o sistema;
- Quais elementos suportarão os principais modos de operação;
- Comunicação entre elementos;
- Verificação se a qualidade está sendo satisfeita;
- Como as funções estarão divididas no sistema;
- Outros, incluindo:
  - Comunicação entre os elementos

## PASSO 6 - DEFINIÇÃO DE INTERFACES

Os seguintes passos são realizados:

- Verificar os requisitos funcionais que envolvem os elementos instanciados no passo anterior;
- Observar informações geradas em um elemento e consumidos em outro;
- Elaborar a documentação da interface de cada elemento;

Nessa etapa, as seguintes decisões e considerações de *design* são atingidas:

- Interfaces externas ao sistema ;
- Interfaces entre as divisões do sistema;
- Interfaces da Infraestrutura ;

## PASSO 7 - VERIFICAR E REFINAR REQUISITOS E FAZER RESTRIÇÕES

- Verificar se as características de projeto atribuídas ao elemento pai foram alocadas para um ou mais elementos filhos na decomposição;
- Traduzir quaisquer responsabilidades atribuídas aos elementos filhos em funções requisitos nacionais para os elementos individuais;
- Refinar os requisitos de atributos de qualidade para elementos filhos individuais.

## PASSO 8 - REPETIR COM O PRÓXIMO ELEMENTO

- Depois de concluir as etapas 1 a 7, você terá uma decomposição do elemento pai em elementos filhos;
- Cada elemento filho é uma coleção de responsabilidades, com seus próprios artefatos;
- Retornar ao processo de decomposição na Etapa 2.

# SAÍDA DO PROCESSO

Como resultado, o ADD produz um projeto de sistema em termos de papéis, responsabilidades, propriedades e relacionamentos entre elementos de software;

O projeto resultante é documentado sob diferentes perspectivas arquiteturais, incluindo as de Módulo, Componente-Conector e Alocação, determinando:

- Como dividir um sistema em seus principais elementos computacionais e de desenvolvimento;
- Quais elementos comporão as estruturas do sistema, o tipo de cada elemento, suas propriedades e relacionamentos;
- Quais interações ocorrerão entre elementos, suas propriedades e mecanismos.



# SMART DECISIONS GAME (SDG)

O principal objetivo do Smart Decisions é ensinar técnicas de tomada de decisão arquitetural usando o Attribute-Driven Design (ADD).

No ADD, o objetivo é decompor um sistema em componentes e fazer escolhas de arquitetura que otimizem esses atributos, como desempenho, segurança, escalabilidade, etc. O Smart Decisions Game simula essa abordagem ao permitir que os jogadores experimentem diferentes decisões e seus impactos em atributos de qualidade do sistema.

## ADD APLICADO AO SDG

**Cenário proposto:** Sistema de Gestão de Veículos Conectados (IoT)

Imagine que você está jogando o Smart Decisions Game e o objetivo do seu projeto é desenvolver um sistema IoT de gestão de veículos conectados. Este sistema será responsável por monitorar e comunicar dados sobre veículos em tempo real, como a localização, o estado do motor, e o consumo de combustível.



# SMART DECISIONS GAME (SDG)

## ADD APLICADO AO SDG

**1 - Identificar Atributos de qualidade prioritários:** São os atributos que guiarão as decisões arquiteturais ao longo do jogo.

- Desempenho
- Segurança
- Escalabilidade
- Atualizações em tempo real

**2 - Organizar os módulos do sistema:** Definir como o sistema será organizado internamente.

Decisão no jogo: O sistema será dividido em três módulos principais:

- Módulo de sensores
- Módulo de processamento central
- Módulo de interface de usuários

Essa divisão garante escalabilidade e modularidade, pois novos veículos podem ser integrados ao sistema com facilidade, e o processamento pode ser distribuído conforme necessário.

# SMART DECISIONS GAME (SDG)

## ADD APLICADO AO SDG

**3 - Perspectiva Componente-Conector:** Como os componentes definidos interagem e se comunicam.

Decisão no Jogo: Para garantir o desempenho em tempo real e a eficiência energética, é decidido que:

- O protocolo MQTT(Message Queuing Telemetry Transport) será usado para comunicação entre os sensores embarcados nos veículos e o servidor central, já que é leve e otimizado para dispositivos com pouca capacidade de processamento.
- O WebSocket será usado entre o servidor central e a interface do usuário, permitindo a visualização de dados em tempo real sem o custo de latência de requisições HTTP convencionais.

*Ao escolher MQTT, por exemplo, os jogadores sacrificam a simplicidade de implementação em troca de eficiência energética e baixo consumo de largura de banda. Essas decisões impactam diretamente atributos de desempenho e escalabilidade, influenciando os jogadores a avaliar os trade-offs de cada escolha.*



# SMART DECISIONS GAME (SDG)

## ADD APLICADO AO SDG

**4- Perspectiva de Alocação:** Como os módulos e componentes do sistema serão alocados.

Decisão no Jogo: Nesta fase, os jogadores decidem onde os componentes serão executados

- Os sensores de veículos serão embarcados nos próprios veículos, com capacidade de armazenar dados localmente em caso de falhas na comunicação.
- O processamento central será executado na nuvem, oferecendo escalabilidade e poder de processamento para lidar com grandes volumes de dados.
- A interface do usuário será acessível por meio de um painel web, que recebe dados diretamente do servidor na nuvem.

*No jogo, os jogadores podem testar diferentes alocações, verificando como elas afetam os atributos de qualidade. A alocação dos sensores nos veículos permite economia de energia, enquanto o uso da nuvem para processamento central garante escalabilidade. O feedback fornecido pelo jogo ajuda os jogadores a ajustar sua estratégia.*

**5- Análise de Trade-offs e Iteração:** Como o ADD é um processo iterativo, o SDG também permite que os jogadores revisem suas escolhas. Ao final de cada rodada, o impacto das decisões é mostrado em termos de como os atributos de qualidade foram atendidos.

**Exemplo de Iteração:** Se o sistema não estiver atendendo aos requisitos de desempenho em tempo real, os jogadores podem decidir mover parte do processamento dos veículos para a borda (edge computing), minimizando a latência da transmissão de dados para a nuvem.

# BENEFÍCIOS

## USUÁRIO

- O ADD garante que os sistemas sejam construídos com o usuário em mente, oferecendo funcionalidades e interfaces intuitivas que atendem às suas necessidades específicas.
- Ao modelar atributos importantes, o sistema se ajusta às mudanças nas necessidades do usuário, proporcionando uma experiência mais satisfatória ao longo do tempo.

## ORGANIZAÇÃO

- A arquitetura flexível e a facilidade de manutenção diminuem os custos de desenvolvimento e operação do sistema ao longo do seu ciclo de vida.
- Sistemas construídos com o ADD tendem a ter maior qualidade, pois são mais robustos, confiáveis e adaptáveis a mudanças.
- Ao oferecer sistemas que atendem às necessidades dos usuários, as organizações aumentam a satisfação dos clientes e a sua fidelidade.

## EQUIPE DE DESENVOLVIMENTO

- A arquitetura flexível e a clara definição de atributos facilitam o desenvolvimento e a manutenção do sistema, aumentando a produtividade da equipe.
- O ADD promove uma comunicação mais clara e objetiva entre os membros da equipe, reduzindo o risco de mal-entendidos e retrabalho.
- A modelagem de atributos auxilia na tomada de decisões mais assertivas durante o desenvolvimento, garantindo que as escolhas técnicas estejam alinhadas com os objetivos do projeto.



**OBRIGADA!**