

# Planejamento vs. Design

Diogo Winck

[diogo.winck@catolicasc.org.br](mailto:diogo.winck@catolicasc.org.br)

## Planejamento

Decidir o caminho que o projeto vai seguir

### Características:

- Define **o que** será feito
- Define **para quem, quando e por quê**
- Cria visão clara de escopo e valor
- Alinha expectativas da equipe e stakeholders

## Design

Decidir como a solução será construída

### Características

- Define **como** o software será estruturado
- Escolhe tecnologias, padrões e arquitetura
- Considera desempenho, escalabilidade e manutenção
- Impacta diretamente na robustez e resiliência

# Aprofundando sobre Planejamento


## 3 pilares

- Visão do Produto (onde)
- Roadmap (o que)
- Execução estratégica (como)

# Visão do Produto

Define o **propósito principal** do produto

Foco em **valor entregue ao usuário final**

 Ajuda a guiar todas as decisões futuras.

# Roadmap

Uma **linha do tempo** com as principais entregas planejadas

📌 Mostra **onde o projeto está indo** no médio/longo prazo

Dois fluxos: Upstream e Downstream

## **Product Backlog**

| Lista viva de requisitos, organizada por prioridade

 **Fonte única de verdade** do que precisa ser feito

| criado pelo **Upstream** executado pelo **Downstream**.

## Upstream (Exploração)

Fase de descoberta, pesquisa e definição **antes do desenvolvimento**

### Características:

- Identificação de oportunidades
- Compreensão de problemas
- Validação de ideias
- Definição de objetivos de produto

 Foco: **eficácia** — fazer a coisa certa



## **Downstream (Entrega)**

Fase de **execução e entrega** após as decisões tomadas

### **Características:**

- Quebra de funcionalidades em tarefas
- Desenvolvimento incremental
- Testes e validação técnica
- Entregas contínuas e feedback

 **Foco: eficiência** — fazer certo a coisa

## Conexão no Roadmap

flowchart LR  
A[Ideia] --> B[Exploração/Upstream]  
B --> C{Decisão}  
C -- Refinamento --> C  
C --> D[Entrega/Downstream]  
D --> E[Feedback]  
E --> A

## Conexão com Roadmap

 O ciclo se repete com novos aprendizados.

 Equipes maduras mantêm o equilíbrio entre pensar e executar continuamente.

## Conexão com Design

Como artefatos resultantes do processo Downstream deve ser um conjunto de especificações técnicas que viabiliza a execução do técnica do produto.

- C4 Model
- Diagramas da UML: Classes, Componentes, Pacotes Sequência, Implantação...
- Casos de testes
- Mockups

# Execução estratégica

## Materialização do Produto

- Daily Planning
- Sprint Planning
- Release Planning

## Daily Planning

Reuniões diárias para **ajustar o plano** de curto prazo

 Foco na **capacidade do time de entregar o que foi comprometido**

Deve-se evitar o formato de: o que fiz, o que farei e impedimentos...

## Sprint Planning

| Define o que será feito na próxima iteração

 Baseado na capacidade da equipe + prioridades do backlog

### 3. Release Planning

| Planeja **quais funcionalidades** serão entregues em cada versão

 Alinha expectativas de entrega com **negócio e equipe**



## O que guia o Design?

- Carga e Escalabilidade
- Assíncrono vs Síncrono
- Deduplicação e Consistência
- Persistência e Modelagem de Dados
- Manutenção e Governança

## Carga e Escalabilidade

### Perguntas para refletir:

- Qual o volume esperado de requisições por segundo?
- Como o sistema se comporta em picos de uso?
- Precisa escalar horizontalmente, verticalmente ou ambas?

 Importância: dimensionar a infraestrutura e definir limites de performance

## **Processamento Assíncrono vs Síncrono**

**Avalie o comportamento esperado:**

- É melhor salvar dados direto ou colocá-los numa fila?
- O que exige resposta imediata?
- Pode haver atraso no processamento?

 Fila: resiliente, desacopla sistemas

 Direto: simples, mas menos tolerante a falhas

## Deduplicação e Consistência

### Garantia de dados únicos:

- Como evitar salvar dados repetidos?
  - Preciso de consistência forte ou eventual?
  - Identifico duplicatas por ID, hash, timestamp...?
- 
- 💡 Consistência forte = mais complexidade
  - 💡 Eventual = mais flexibilidade, mas cuidado com duplicatas

## Persistência e Modelagem de Dados

Escolhendo a tecnologia de dados certa:

- Qual banco é mais adequado para históricos com data e métrica?

Tipo de Dado	Banco Recomendado
Relacional	PostgreSQL, MySQL
Temporal	TimescaleDB, InfluxDB
Documentos	MongoDB, DynamoDB

- Como estruturar os dados para performance?

## Manutenção e Governança

### Automatizando a limpeza de dados:

- Volume de dados é um problema?
- Deve-se limpar a base?
- O que define um registro inativo? e excluído?
  - Sem requisições?
  - Sem atualizações?

## Exercício 1

## Trabalho 2