

# BANCO PAN JAVA

09.02.23/1

## 1. SISTEMA Desktop vs 2. SISTEMA WEB

↳ Muito utilizado no passado.  
↳ Excel

↳ Novo  
↳ Excel 365  
ou google sheets

## 3. SISTEMA MOBILE

### PROFISSIONAIS DE UM DESENV. DE SOFTWARE

#### → UX Design (USER EXPERIENCE)

↳ FOCO NA MELHORIA DA USABILIDADE DO USUÁRIO DO MEU PROJ.

- Utiliza plataformas para criação de protótipos
- Faz muitas pesquisas

#### → UI Design (USER INTERFACE)

↳ Focado em cores, tipografia, microinterações, etc.

↳ Materializar o que o UX Design pensou

↳ MAS NÃO É O DESENVOLVEDOR MUITAS VEZES.

- Utiliza Photoshop por exemplo.

COMPORTAMENTO

APARÊNCIA

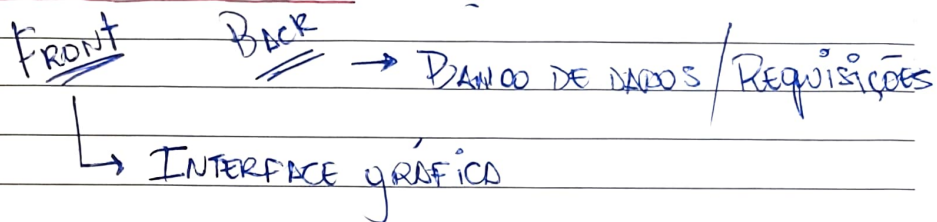
UX

UI

Pesquisa, protótipo ...

Design, tipografia, cores, ...

## Modelo Cliente - Servidor



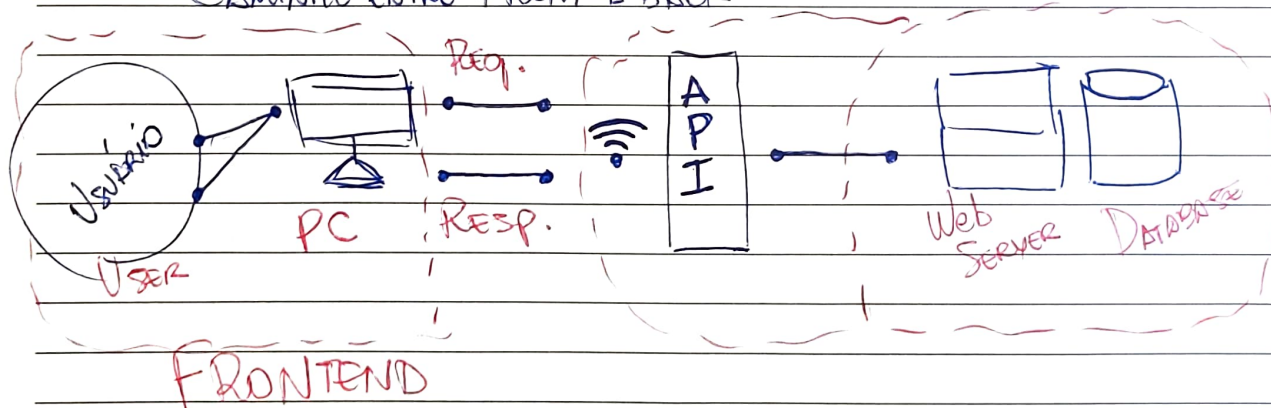
\* TDE - Editor de código

### \* FRAMEWORKS

- Soluções pré-definidas
- Facilitador
- Angular (HTML, CSS & JS) - FRONT
- Springboot (JAV) - BACK
- Vue (JavaScript)

### \* API

- Caminho entre aplicações
- Caminho entre FRONT e BACK



09.02.23

→ Full Stack - top! Sabe back e front.

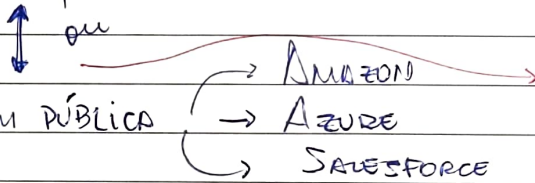
→ QA - Quality Assurance

- TESTAR A QUALIDADE DO PROJETO
- TESTAR O RETORNO E USABILIDADE DO USUÁRIO
- CONHECIMENTO DO NEGÓCIO

10.02.23

→ INFRAESTRUTURA

- SEGURANÇA DA TECH. E INF.
- INFRAESTRUTURA LOCAL
- NUVEM PRIVADA

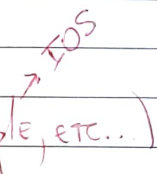
→ CLOUD 

- NUVEM PÚBLICA

Cloud Computing  
• RESPONSÁVEL PELA  
INTRA E ORGANIZAÇÃO  
DA UTILIZAÇÃO DA  
NUVEM

+ BARATO

+ FACILIDADE DE CONTRATAÇÃO

→ DESENVOLVIMENTO MOBILE (Android, Apple, etc...) 

Android → JAVA / KOTLIN

IOS → SWIFT

HÍBRIDO → FLUTTER / REACT NATIVE / IONIC

## TRABALHANDO EM EQUIPES ÁGEIS

(TREMULDA)      (QUEBRADO)

Rápido ≠ Ágil → EFICIÊNCIA

                                ↓

                                ADAPTAÇÃO

                                ↑

                                RESILIÊNCIA

↳ Precisamos muito mais ter um time ágil do que VELOZ.

MANIFESTO Agil → 2021

Os 4 valores

\* 12 principios

# 1. INDIVÍDUOS E INTERAÇÕES

2. SOFTWARE EM FUNCIONAMENTO  
MAIS QUE DOCUMENTAÇÃO ABRANGENTE

3. Responder a mudanças  
mas que seguir um plano

4. COLABORAÇÃO COM O CLIENTE  
MAIS QUE NEGOCIAÇÃO DE CONTRATOS

### \* CARACTERÍSTICAS DE UM TIME Ágil

- ALINHADO COM O CLIENTE E STAKEHOLDERS
- AUTO-ORGANIZADO E RESPONSÁVEL
- MULTIDISCIPLINAR
- ENTREGA VALOR CONTINUAMENTE
- ESTÁ SEMPRE APRENDENDO
- MELHORIA CONTÍNUA
- POSSUI MÉTRICAS E METAS CLARAS
- UM DO

10.02.23

# GERENCIAMENTO DE PROJETOS

## Processo

- Recorrente
- Contínuo
- Replicável

## Projeto

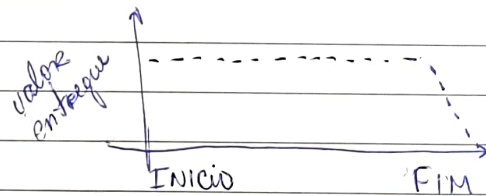
- Temporário
- Exclusivo

## Ágil vs. TRADICIONAL

↳ MÉTODO CASCATA (WATERFALL)

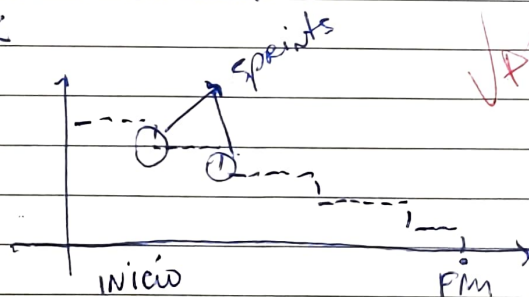
→ ANÁLISE → PROJETO → IMPLEMENTAÇÃO ...  
... → TESTE → ENTREGA/MANUTENÇÃO

### • Ciclo contínuo



VALOR ENTREGUE  
NO FINAL

- Feito por etapas (sprints)
- ITERATIVO E INCREMENTAL
- FEEDBACK



VALOR ENTREGUE  
CONTINUAMENTE



# MÉTODOS ÁGILS

"FRAMEWORK"

BASES DO SCRUM →

EMPIRISMO

LEARN Thinking

PILARES

→ ciclo PDCA

→ MELHORIA CONTÍNUA  
E EVITAR DESPÉDICIOS

• TRANSPARÊNCIA

→ TODOS SADEM DE TUDO SOBRE O PROCESSO

• INSPEÇÃO: → PENSAR

→ POSSIBILITA A ADAPTAÇÃO

• ADAPTAÇÃO:

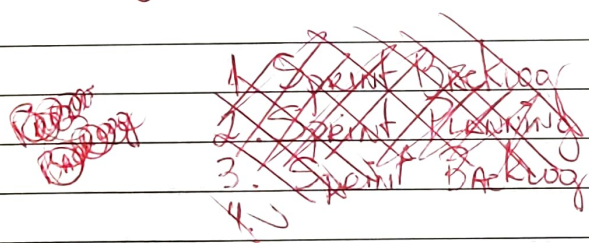
METODOLOGIA  
(RECEITA)  
→ DETALHADO

x

FRAMEWORK  
(ESQUELETO)

→ NÃO FAÇA O "COMO?"

## SCRUM Framework



Backlog → Sprint Planning

→ SPRINT (TEAM → SM → PO → Devs) → INCREMENT → Sprint Review

RETO

KANBAN

→ MÉTODO

→ CARTÕES

→ GESTÃO DE TAREFAS

## → OBJECTIVES KEY RESULTS

OKR → Simplificar a forma de ENCARAR OBJ. ESTRATÉGICOS

→ Descrições Qualitativas de objetivos

↳ OBJETIVOS PRECISAM SER:

- CLARO
- INSPIRADOR
- DESAFIADOR
- ALINHADO COM A MISSÃO DO NEGÓCIO

PARA CADA OBJETIVO TEMOS KR'S (METAS) QUE PRECISAM SER:

→ BASEADA EM KPI'S

→ QUANTITATIVAS

→ SMART

- ↳ ESPECÍFICAS
- ↳ MENSURÁVEIS
- ↳ ALCANÇÁVEIS
- ↳ RELEVANTES
- ↳ TEMPORAIS

↳ DE 2 A 5 KR'S POR OBJETIVOS

↳ NORMAMENTE OBJETIVOS ANUAIS, SEMESTRAIS, TRIM., ETC.

## BENEFÍCIOS

## → FERRAMENTAS


- AGILIDADE
- COOPERAÇÃO
- TRANSPARÊNCIA
- MONITORAMENTO CONTÍNUO


- Trello
- Notion
- Miro
- Easypretao
- Clickup


## JAVA BÁSICO

```
public class MinhaClass {  
    // código  
}
```


## OPERADOR LÓGICOS

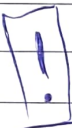
→ Conjuncão (AND) 

→ Disjunção (OR) 

→ Disjunção Exclusiva (XOR) 

V	V	-	V
V	F	-	F
F	V	-	F
F	F	-	V

→ Negação (NOT) 

→ Inversão 

## CONTROLE DE FLUXO

- Decisão: If, Else, Switch
- Repetição: For, While, Do While
- Interrupção: Break, Continue, Return