

# Desenvolvimento para Sistemas Web

Código da turma no classroom: **dtdzf7u**

Aulas: Terça - 20:20 e Quarta - 18:30



# Para aprofundar!

Leitura para a próxima aula (Terça-Feira)!

Artigo: Desenvolvimento de sistemas Web orientado a reuso com Python, Django e Bootstrap

Trazer **informações relevantes e impressões**

# Para aprofundar!

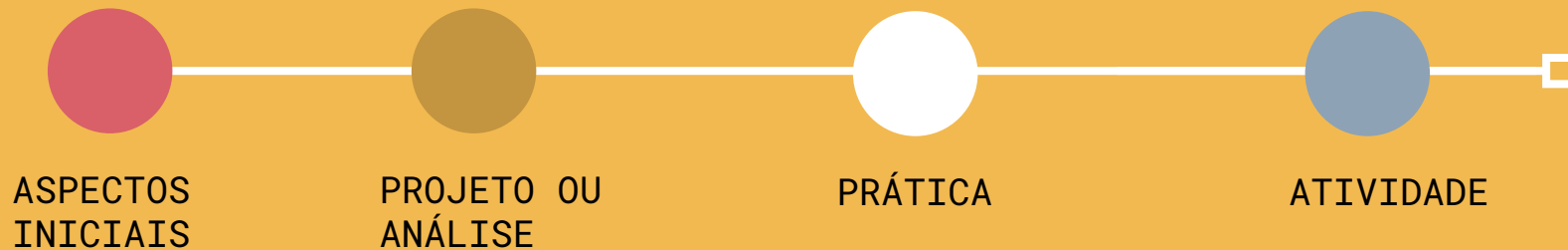
Acessem os links na caixinha.

Acessar a documentação do  
Github: ver aspectos  
gerais, recursos e  
instruções iniciais

Solicitar **GitHub**  
**Student Developer**  
**Pack**

# Etapas do Desenvolvimento Web

# O QUE TEREMOS NA AULA DE HOJE!



# Iniciando o nosso projeto!

O desenvolvimento de sistemas é um processo que ocorre em algumas etapas, como projeto ou análise, codificação, testes, implantação, treinamento e manutenção.

A definição das etapas pode variar de caso para caso, mas de uma forma geral o processo de desenvolvimento está relacionado às seguintes etapas.

# Etapa - Projeto ou Análise



O projeto é a parte mais importante do sistema. Nele é definido **o que, como, quando e porque será feito.**

Diversas metodologias ou paradigmas de projeto podem ser usados, creio que atualmente o paradigma mais adequado seja a **Análise Orientada a Objeto.**

No projeto são realizados diversos tipos de **análises, prototipação, desenhos de cenários, diagramas e outras representações** para tentar aproximar ao máximo a **expectativa e o resultado final** do sistema.

# Etapa - Codificação

A codificação é o **desenvolvimento** propriamente dito.

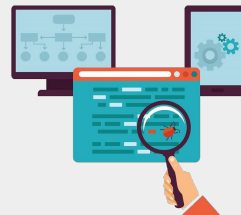
Aqui são escritas linhas de códigos que interligam o sistema com bancos de dados e outros dispositivos a fim de produzir o **resultado** que **foi especificado** no projeto.



A codificação é uma atividade subordinada no projeto.



# Etapa - Testes



A etapa de testes é importante para verificar a existência de falhas ou se o sistema está se comportando dentro dos princípios do projeto.

A realização de testes não deve ser feita de qualquer maneira ou por qualquer pessoa. Para obter um bom resultado, o **ideal é estabelecer procedimentos para testes**, que deverá incluir:

- O que deve ser testado;
- Qual dado deverá ser inserido;
- Qual o perfil da pessoa que irá testar: amador, experiente, usuário, etc;
- Se o procedimento deverá ser repetido com variações de dados;
- Entre outros.

# Etapa - Implantação ou Produção



Se aprovado nos testes, o sistema deverá ser implantado, isto é, **colocar em produção**.

Essa etapa implica movimentar pessoas, hardwares, softwares e muitas vezes com a empresa em pleno funcionamento.

Dependendo do tipo e tamanho do sistema é necessário determinar momentos estratégicos para a implantação a fim de provocar menos impacto nos procedimentos da empresa/cliente.

# Etapa - Manutenção



A última etapa do processo de desenvolvimento de sistemas é a **manutenção**.

Ela inicia logo após a implantação e não terá fim, ou seja, existirá enquanto o sistema existir.

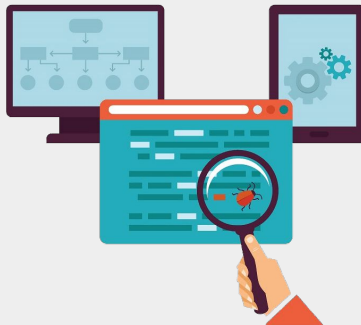
A manutenção é uma etapa de muitas controvérsias, por isso é importante existir um **documento que regulamente o que é manutenção**, quais as responsabilidades de clientes, usuários e desenvolvedores, bem como os custos envolvidos nessas atividades.



Projeto ou Análise



Codificação



Testes



Produção



Manutenção

# Etapa de Projeto ou Análise

# Projeto ou Análise

Defina minuciosamente os **aspectos do sistema** antes de mergulhar no código.

Simplesmente porque é uma forma **mais funcional e prática de testar** a sua ideia.

Afinal de contas, não é sempre que você poderá garantir a **usabilidade** do que tem em mente. Além disso, caso decida programar antes de projetar, poderá gastar um **tempo** considerável se a implementação frustrar as suas expectativas.

## Introdução

Uma breve descrição da motivação para o desenvolvimento do projeto, alguns benefícios, problema atual, proposta de solução.

## Visão Geral do Produto

Descreve o objetivo do sistema, suas respectivas funcionalidades, a visão da solução fornece informações sobre um possível caminho para solucionar o problema já identificado.

## Premissas e Restrições

Premissas e Restrições são hipóteses e limitações.

## Visão Geral do Produto

Descreve o objetivo do sistema, suas respectivas funcionalidades, a visão da solução fornece informações sobre um possível caminho para solucionar o problema já identificado.



## Requisitos Funcionais

São os requisitos funcionais do sistema que o sistema pode oferecer, expressam o comportamento de um software, são as necessidades apontadas pelo cliente, ou seja, o que ele quer que o sistema faça.

## Requisitos Não Funcionais

Mapeiam os aspectos qualitativo de um software, por exemplo a performance (tempo de resposta), segurança, comunicabilidade entre outros são as características e aspectos internos do sistema, envolvendo especificamente a parte técnica.

## Layout

Layout tem como seus componentes a área de design ou formato de página e as margens, deve ser bem fundamentado pelo conteúdo do trabalho e pela perspectiva do cliente ou pela interatividade com o usuário

## Prototipagem

Versão não funcional do sistema!

## O que é prototipagem?

A prototipagem é um termo técnico que explica a prática de prototipar, ou seja, **criar um protótipo de algo que deseja produzir.**

A ideia é elaborar uma versão inicial, um **esqueleto** por assim dizer, para usá-lo como material de apresentação no pitch da sua ideia para a parte interessada.

## O que é prototipagem?

O objetivo dessa etapa é criar um **modelo do produto final**, que não precisa apresentar a qualidade técnica e o polimento do app finalizado.

Na realidade, basta que essa versão transmita a ideia de **usabilidade**, **estilo** e **propósito** da sua solução.

## O que é prototipagem?

O objetivo do protótipo é apenas **simular um modelo real** da sua ideia, sendo uma ponte entre o que você tem em mente e a representação que você consegue criar disso no seu protótipo.

## Qual a importância da prototipagem?

Com um protótipo, que pode ter diferentes níveis de fidelidade, você já consegue uma boa noção da qualidade e do engajamento gerado pela sua ideia.

Nesse sentido, ela pode ser um estilo de interface do usuário, um novo posicionamento de botões, formas de interações ou, até mesmo, o conceito.

## Quais as principais vantagens?

### Testes controlados

Principalmente  
**Usabilidade** (experiência do usuário) e  
**Funcionalidade** (se realmente resolve o problema)

### Feedbacks valiosos

Aproveita a oportunidade para absorver os comentários de quem testou a ferramenta.

## Quais as principais vantagens?

### Economia de recursos

É uma forma muito inteligente de otimizar sua produtividade, aplicando dedicação em doses e momentos necessários para a alavancagem da sua ideia.

Depois que a aplicação passar pela bateria de testes, a prova de fogo da opinião dos primeiros usuários, aí sim você considera os próximos passos, poupando tempo e tornando sua atuação muito mais inteligente e ágil.



## Quais as variações de fidelidade?

Fidelidade é o nível de semelhança conceitual, estética e funcional entre a ideia e o produto final.

### Fidelidade Baixa

Esse é o tipo de protótipo mais fácil e prático de ser desenhado, bastando uma caneta e um bloco de notas. Literalmente é um **rascunho** da ideia.

## Quais as variações de fidelidade?

### Fidelidade Média

Solução intermediária na prototipagem. Nesse cenário, o mockup não é idêntico ao produto final, mas também não é tão rudimentar como o primeiro rabisco.

### Fidelidade Alta

Replica o comportamento da **solução final com fluidez**, apresentando os elementos de **design** do produto final, bem como as **funcionalidades** esperadas na versão que será lançada no mercado.

# Praticando os conceitos de Projeto ou Análise

Considerando os conceitos estudados nesta aula, defina em seu projeto:

## Subir no Github

Arquivo \*.md

Modelo no Repositório

- Introdução
- Visão Geral do Projeto
- Premissas e Restrições
- Requisitos Funcionais e Requisitos Não Funcionais
- Layout/Prototipagem

# Desenvolvimento para Sistemas Web

[sarahsoares.com.br/](http://sarahsoares.com.br/)

