

---

---

# Desenvolvimento WEB

-

Full Stack Completo: Java + React

---

---

# Introdução a Arquitetura de Software



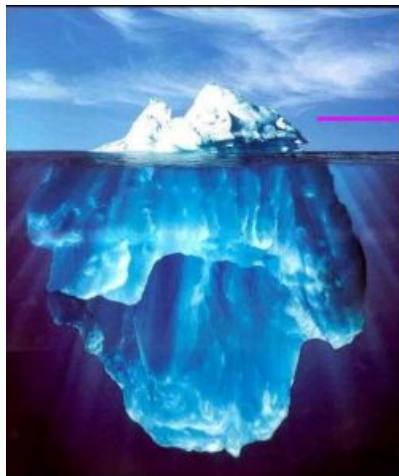
# Introdução a Arquitetura de Software

## O que é Arquitetura de Software?

- O projeto de arquitetura está preocupado com a compreensão de como um sistema deve ser organizado e com a estrutura geral desse sistema.
- De forma geral, arquitetura é uma divisão prudente do “todo” em partes, com relações específicas entre as partes.
- Estrutura de componentes de um programa/sistema, os relacionamentos entre esses componentes, os princípios e diretrizes que governam os projetos e a evolução dos softwares.  
(SEI - Software Engineering Institute)
- O resultado do processo de projeto de arquitetura é um modelo de arquitetura que descreve como o sistema está organizado em um conjunto de componentes de comunicação.

# Introdução a Arquitetura de Software

## O que é Arquitetura de Software?



Comportamento, Interação

Implementação, Algoritmos e Estrutura  
De Dados

A Arquitetura é um dos produtos de  
um projeto (*design*) de software

# Introdução a Arquitetura de Software

## Influências na arquitetura

- Interessados (stakeholders) no sistema;
- Organização onde é desenvolvida;
- Experiência e conhecimento do arquiteto;
- Ambiente técnico.

# Introdução a Arquitetura de Software

## Então, o que é uma arquitetura?

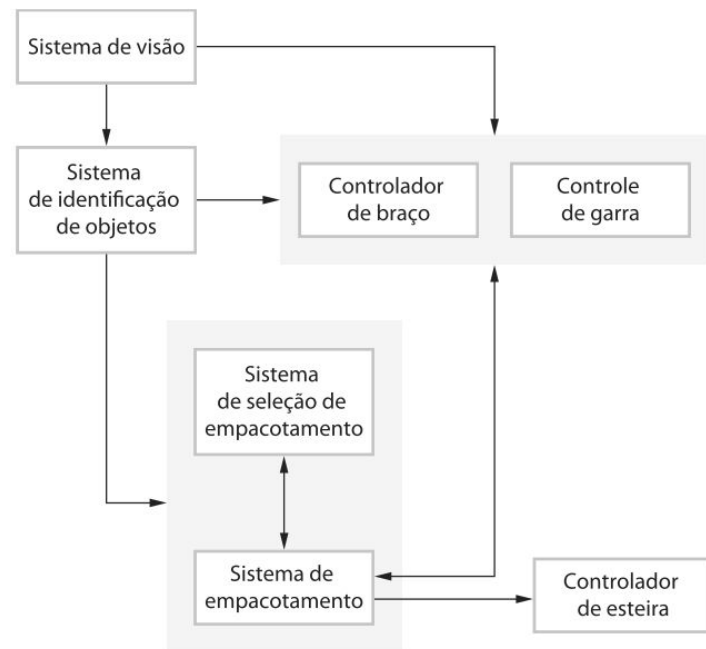
- Uma arquitetura deve definir como os elementos se relacionam uns com os outros;
- Uma arquitetura deve ter diferentes estruturas (visões);
- Uma arquitetura deve descrever o comportamento dos elementos que fazem parte da mesma.

# Introdução a Arquitetura de Software

## Formas de representar a arquitetura

- Em geral, as arquiteturas de sistema são modeladas por meio de diagramas de blocos simples.
- No diagrama, cada caixa representa um componente.
- Caixas dentro de caixas indicam que o componente foi decomposto em sub componentes.
- As setas significam que os dados e/ou sinais de controle são passados de um componente a outro na direção das setas.

A arquitetura de um sistema de controle robotizado de empacotamento

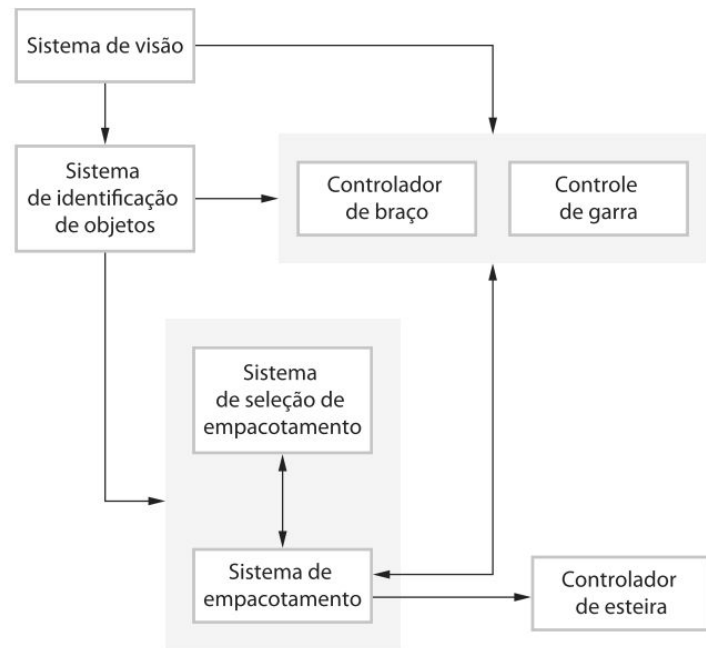


# Introdução a Arquitetura de Software

## Formas de representar a arquitetura

- Diagramas de blocos apresentam uma imagem de alto nível do sistema, de forma que pessoas de diferentes disciplinas possam compreender facilmente;
- No entanto, muitos profissionais não consideram esse como o melhor diagrama para representação de arquitetura.

A arquitetura de um sistema de controle robotizado de empacotamento





# Introdução a Arquitetura de Software

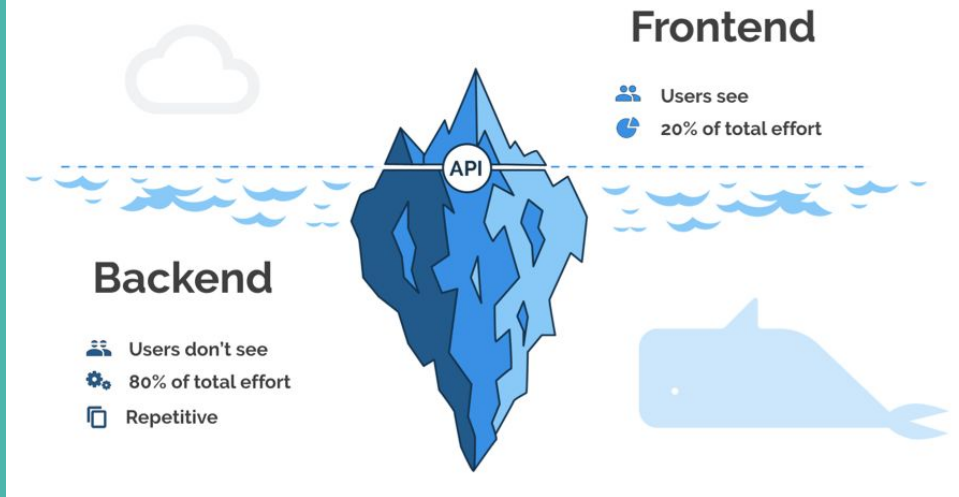
## Importância da Arquitetura

- Instrumento de comunicação entre os interessados no projeto;
- Manifesta antecipadamente decisões de projeto;
- Serve também para transferência de conhecimento do sistema.

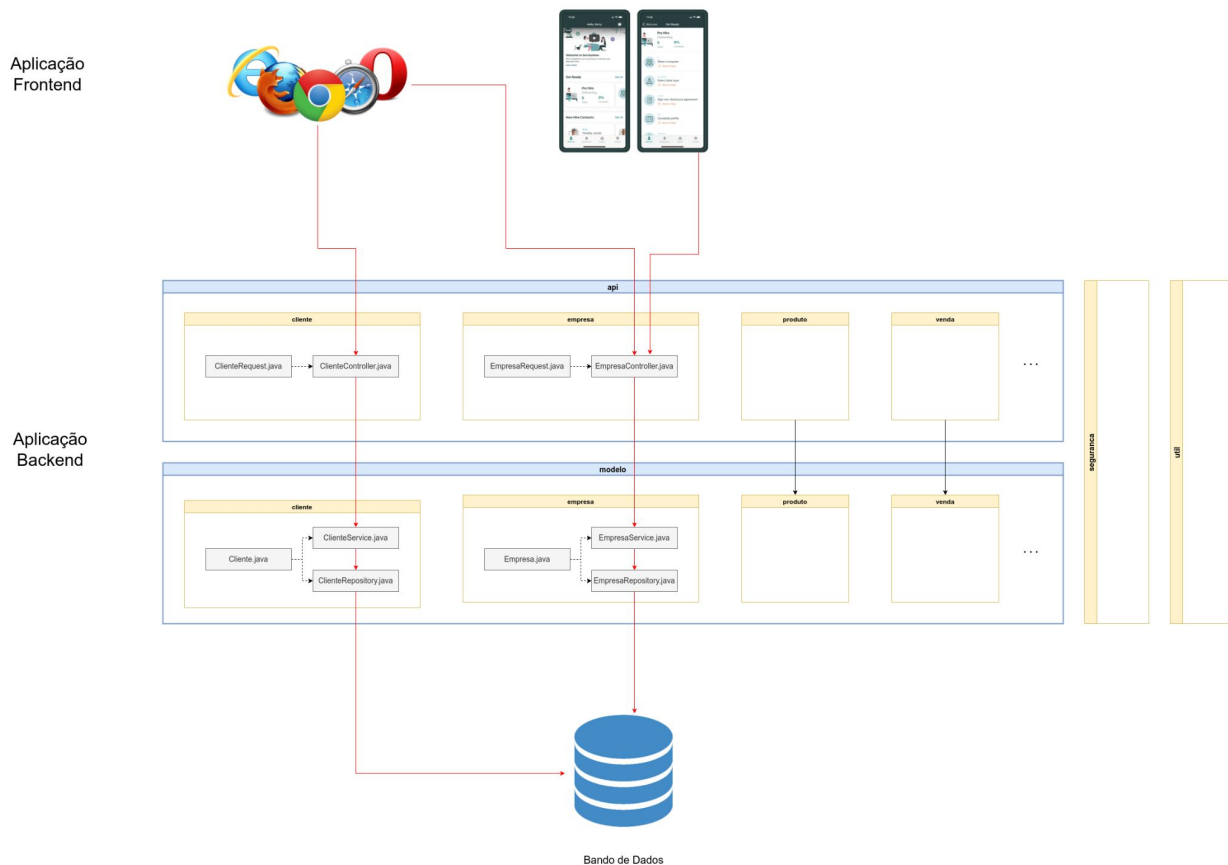
Uma arquitetura mal feita pode  
destruir um projeto

Uma arquitetura ideal não  
garante o sucesso de um projeto

# Iniciando a Implementação do Projeto do Back-end



# Estrutura de Camadas e Pacotes do Projeto



Ver imagem no Google Classroom

# Implementando o Projeto

## 1) Crie a seguinte estrutura de pacotes no projeto:

- `br.com.ifpe.oxefood`
  - `api`
  - `config`
  - `modelo`
  - `util`
    - `entity`
    - `exception`

# Implementando o Projeto

## 2) Crie as classes em **vermelho** abaixo:

- `br.com.ifpe.oxefood`
  - `api`
  - `config`
  - `modelo`
  - `util`
    - `entity`
      - **`EntidadeAuditavel.java`**
      - **`EntidadeNegocio.java`**
    - `exception`

# Implementando o Projeto

## 3) Implemente a classe `EntidadeNegocio`:

```
package br.com.ifpe.oxefood.util.entity;

import java.io.Serializable;

import lombok.EqualsAndHashCode;
import lombok.Getter;
import lombok.Setter;

@Getter
@Setter
@EqualsAndHashCode(of = { "id" })
public abstract class EntidadeNegocio implements Serializable {

    private Long id;

    private Boolean habilitado;

}
```

# Implementando o Projeto

## 4) Implemente a classe `EntidadeAuditavel`:

```
package br.com.ifpe.oxefood.util.entity;

import java.time.LocalDate;

import lombok.Getter;
import lombok.Setter;

@Getter
@Setter
public abstract class EntidadeAuditavel extends EntidadeNegocio {

    private Long versao;

    private LocalDate dataCriacao;

    private LocalDate dataUltimaModificacao;

    private Long criadoPor; // Id do usuário que o criou

    private Long ultimaModificacaoPor; // Id do usuário que fez a última alteração

}
```

# Implementando o Projeto

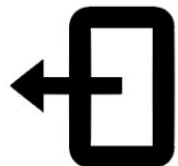
In case of fire



1. `git commit`



2. `git push`



3. `leave building`



# Dúvidas



The image features a white background with two large, solid green abstract shapes. One shape is a semi-circle on the left side, and the other is a more complex, organic shape on the right side. Centered between these shapes is the text "Obrigado !".

Obrigado !