

# Desenvolvimento com frameworks e componentes

# Apresentações

# Objetivos

**Entender a demanda e os conceitos que originaram as "aplicações corporativas"**

# Objetivos

**Perceber a importância de  
componentização e quais as  
consequências dessa prática**

# Objetivos

**Entender o que são containers e como eles interagem com os componentes**

# Objetivos

**Conhecer a especificação JEE e seus principais elementos**

# Objetivos

**Implementar exemplos das principais estruturas da arquitetura JEE: session beans, MDBs e entidades.**

# Objetivos

**Conhecer a proposta do framework Spring  
e seus principais elementos**



# Objetivos

**Implementar exemplos de aplicações  
utilizando Spring.**

# Objetivos

**E o mais importante...**

# Divertir-se!



# Informações

**As informações sobre o curso estarão disponíveis no meu repositório GitHub**

**<https://github.com/michelav/fa7-frameworks>**

**A avaliação será feita por meio de um trabalho a ser definido oportunamente**

# Boom dos anos 80 e 90...



**Aplicações monolíticas e  
"departamentalizadas" proliferavam nas  
grandes empresas**

# Principais características

## Coesão e encapsulamento...

Código que trata dos procedimentos de negócio é implementado em conjunto com aqueles que realizam tarefas de "infraestrutura".

# Principais características

## Reuso...

Por meio de dados e comportamento duplicados.



# Principais características

## Escalabilidade...

Vertical, por meio de aumento de hardware.

# Principais características

## Interoperabilidade...

Quase inexistente, prevalecendo o compartilhamento de dados. Ex.:

- Troca de arquivos
- visões em BD

Aplicações eram verdadeiras "ilhas".

# Concluindo...



**E agora... Como escalar e  
interoperar?**

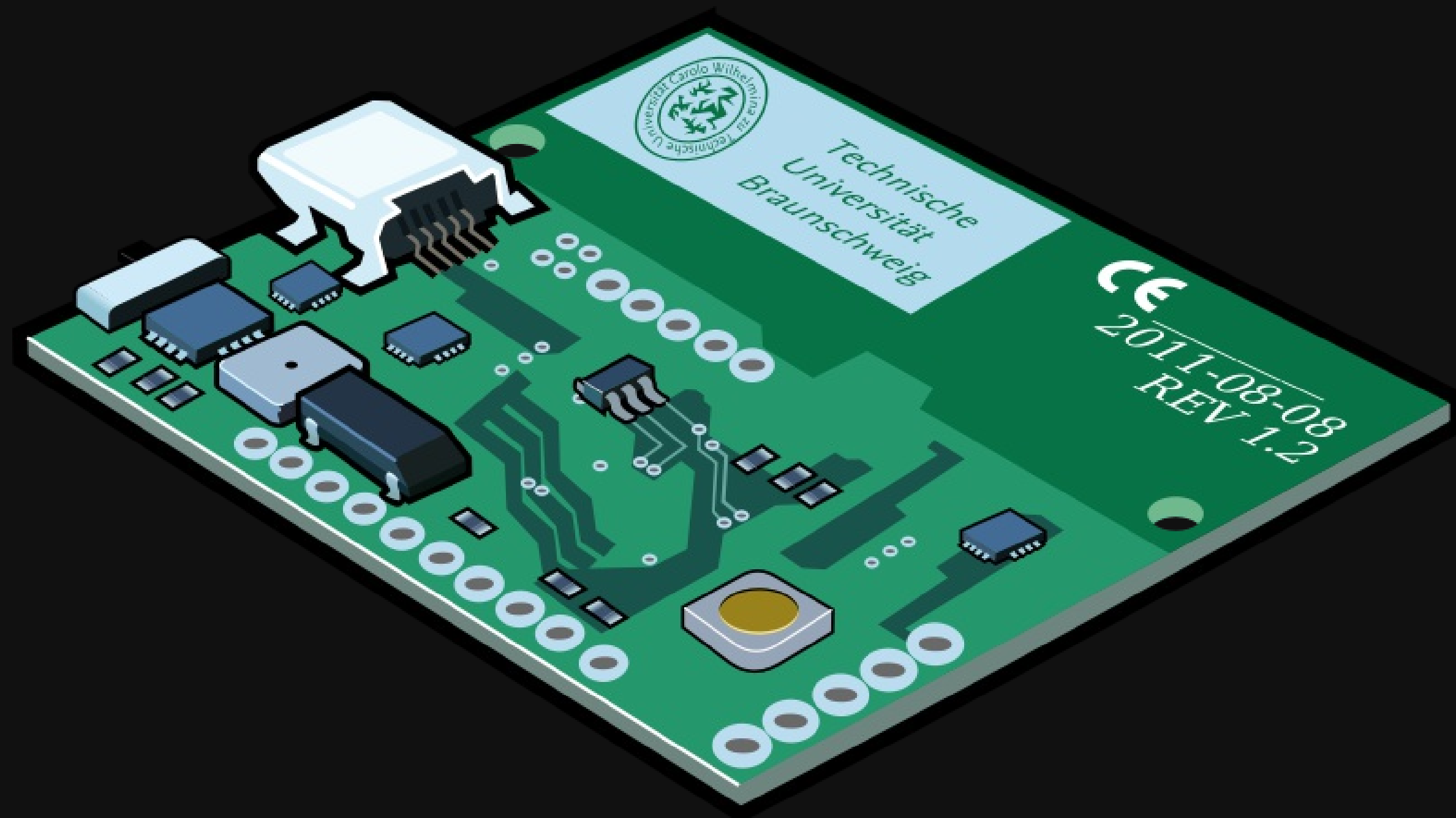
# Componentização



# Componente

## Definição

Artefato (ou conjunto de artefatos) que forma uma unidade de **comportamento** e **interface** bem definidos e pode ser implantado, testado, **composto** e **reusado** de forma independente.



# Componentes de software demandam

Chamada Remota • Balanceamento • Integração • Controle Transacional • Auditoria • Threading • Assincronia • Pooling • Segurança • Caching



EJB dCOM ActiveX Corba OLE Servlets Spring

# Recapitulando...

## Proliferação de aplicações

O crescimento da TI nas décadas de 80 e 90 alavancaram a criação de aplicações monolíticas. Os procedimentos, antes manuais, foram automatizados na forma de sistemas.

# Recapitulando...

## Críticas ao modelo arquitetural

Novos requisitos não funcionais foram propostos. Escalabilidade e integração plena não era mais um luxo e sim necessidade básica. Dessa forma, surgiu a necessidade de uma nova arquitetura que promovesse o reuso, integração e escalabilidade.

# Recapitulando...

## Novo modelo arquitetural

A ideia de agrupar código / comportamento / procedimentos em componentes permitiu tratar esses artefatos de forma separada, distribuindo-os, reusando-os e escalando-os de forma a atender os requisitos impostos pelos usuários das aplicações.

# Recapitulando...

## E o que Java tem com isso?!

A plataforma JEE tornou-se a proposta adequada para desenvolvimento de aplicações corporativas em Java.

