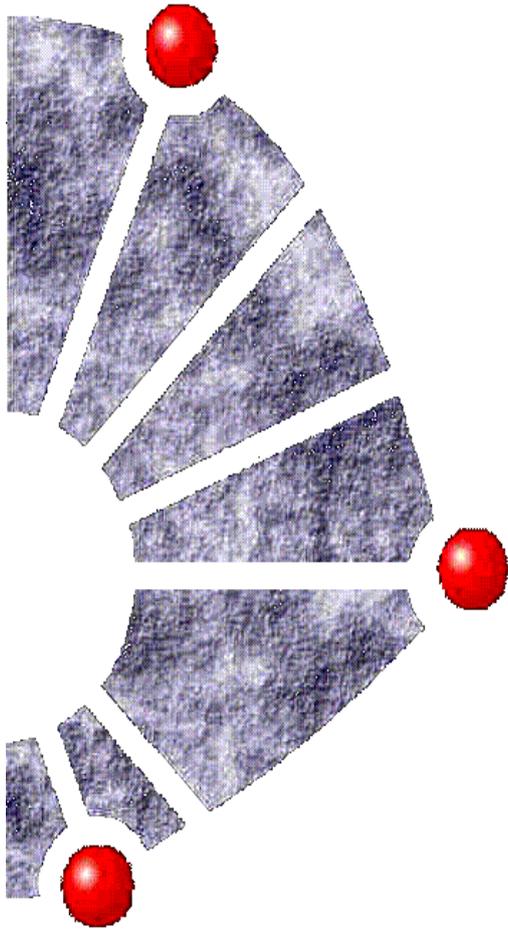


# EA976 – Engenharia de Software

---



## **AULA 26**

### **Componentes, Frameworks e Design Patterns**



# Componentes

---

- O que é um Componente ?
  - Uma tecnologia ou uma abstração ?
  - O que caracteriza um componente ?
  - O que é a interface de um componente ?
  - Como posso representar uma interface em UML ?
  - Qual a relação entre a interface de um componente e uma interface Java ?
  - Qual a relação entre um componente e um objeto ?
  - Dê exemplos de tecnologias de componentes ?
    - Qual a relação entre essas tecnologias ?



# Frameworks

---

- O que é um Framework ?
  - Qual a relação entre um Framework e um Componente ?
  - Qual a relação entre um Framework e uma Biblioteca de Funções ? E uma API ?
  - Qual diferença entre um Framework e Design Patterns ?
  - Como diferenciar um design orientado a Frameworks de um design orientado a Componentes em um diagrama UML ?
  - Como decidir por um design orientado a Frameworks ou um design orientado a Componentes ?



# Design Patterns

---

- O que é um Pattern ?
  - Qual é a origem desse conceito ?
  - Como um pattern é codificado ?
  - O conceito de pattern é aplicado somente ao design ? Existem outros tipos de patterns ?
  - É possível "inventar" um pattern ? Por quê ?
  - Quais os patterns mais importantes ? Quais as fontes de bibliografia para estudar patterns ?
  - O que é um "anti-pattern" ?
  - Por que o conceito de pattern é fundamental para um bom design ?



# GRASP Patterns

---

- O que são os GRASP Patterns ?
  - O que é "atribuir responsabilidade" a um objeto ?
  - Quando os GRASP patterns são importantes ?
  - Quem particularmente se beneficia do uso dos GRASP Patterns ?
  - Quais os dois tipos de responsabilidades que um objeto pode ter ?
    - Como um objeto pode exercer essas responsabilidades em cada caso ?
  - Quais são os 5 GRASP Patterns ?
  - Por que devemos conceber o cerne de uma arquitetura OO como uma "equipe" de objetos ?
    - Como montar essa "equipe" ?



# GRASP Patterns

---

- Como é o pattern Expert ?
  - Quais os benefícios desse pattern ?
  - Como utilizar o Expert ?
- Como é o pattern Creator ?
  - Quais os benefícios desse pattern ?
  - Como utilizar o Creator ?
  - Qual a relação do Creator com o Factory (GoF) ?
- Como é o pattern Low Coupling ?
  - Quais os benefícios desse pattern ?
  - Como utilizar o Low Coupling ?



# GRASP Patterns

---

- Como é o pattern High Cohesion ?
  - Quais os benefícios desse pattern ?
  - Como utilizar o High Cohesion ?
- Como é o pattern Controller ?
  - Quais os benefícios desse pattern ?
  - Como utilizar o Controller ?
  - Como usar o Controller em Java ?
- Os GRASP Patterns devem ser utilizados isoladamente ou podem ser combinados ?



# GoF Patterns

---

- O que são os GoF Patterns ?
  - Como podemos organizar os GoF Patterns, para efeito de classificação ?
  - Qual a importância dos GoF patterns ?
  - Explique algum(s) GoF pattern(s) que você achou mais significativo
    - Explique o porquê !
  - Como os GoF patterns podem auxiliar o seu design ?
  - Existem outros grupos de patterns como os GoF patterns ?