

Recomendador de Objetos de Aprendizagem Baseado em Competências

Recoacomp

Rodrigo Freitas Leite

UFRGS-NUTED

4 de Outubro de 2013

Sumário

1 Concepção

- Análise de Requisitos
- Caso de Uso

2 Projeto e Desenvolvimento

- Padrão Arquitetural
- MVC
- Modelagem de Entidades

3 Conclusão

Conceitos

Objetos de Aprendizagem (OA)

Objetos de aprendizagem são qualquer entidade, digital ou não digital, que possa ser referenciada durante o aprendizado.

Competência

Competências são um conjunto de elementos compostos pelos Conhecimentos, Habilidades e Atitudes (CHA). Tal conjunto é estruturado e mobilizado em um contexto determinado com o intuito de solucionar um problema e/ou lidar com uma situação nova.

Sistema de Recomendação

Sistema que visa auxiliar o usuário na busca e seleção de um conteúdo, funcionando como um filtro de informação. Assim, o usuário terá como resultado OA mais relevantes, conforme utiliza e alimenta o sistema com novas informações, seja do perfil, seja das pesquisas realizadas.

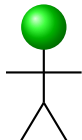
Objetivo

Propósito

O recomendador deve fornecer Objetos de Aprendizagem que supram a necessidade das competências do usuário, ou seja, o recomendador ajuda o usuário a construir suas competências.

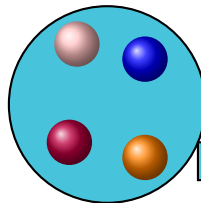
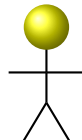
Cenário Básico

Reflexão	
Conhecimento	7
Habilidade	2
Atitude	3



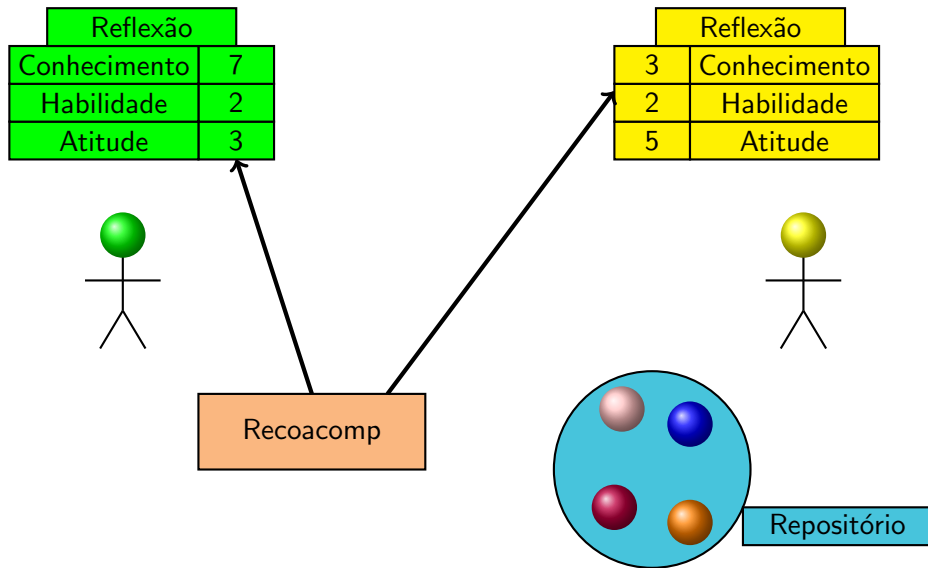
Recoacomp

Reflexão	
3	Conhecimento
2	Habilidade
5	Atitude



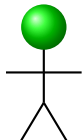
Repositório

Análise das Competências dos Usuários

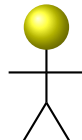


Busca no Repositório por Objetos de Aprendizagem

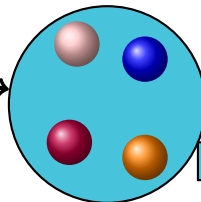
Reflexão	
Conhecimento	7
Habilidade	2
Atitude	3



Reflexão	
3	Conhecimento
2	Habilidade
5	Atitude

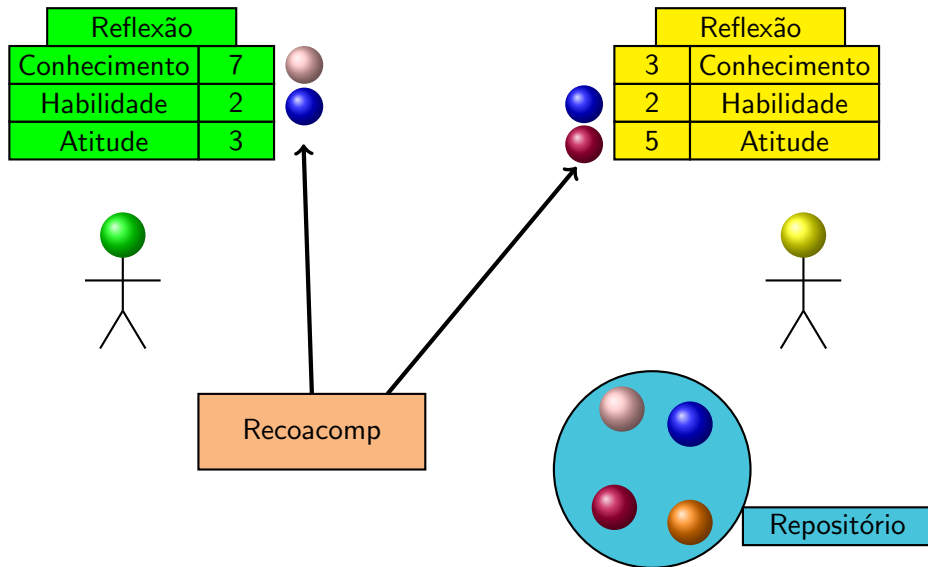


Recoacomp



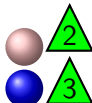
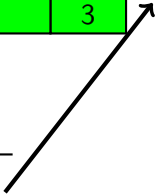
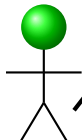
Repositório

Recomendação de Objetos de Aprendizagem



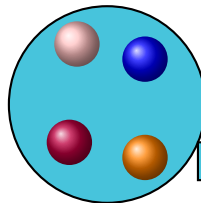
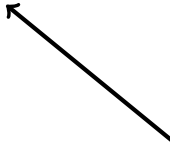
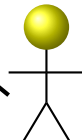
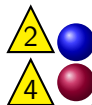
Avaliação dos Objetos de Aprendizagem pelos Usuários

Reflexão	
Conhecimento	7
Habilidade	2
Atitude	3



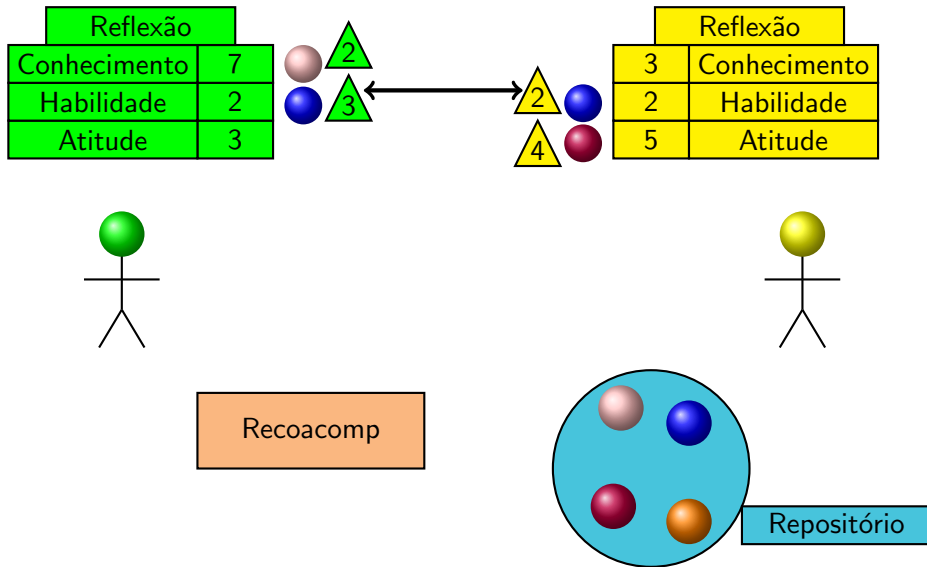
Recoacomp

Reflexão	
3	Conhecimento
2	Habilidade
5	Atitude

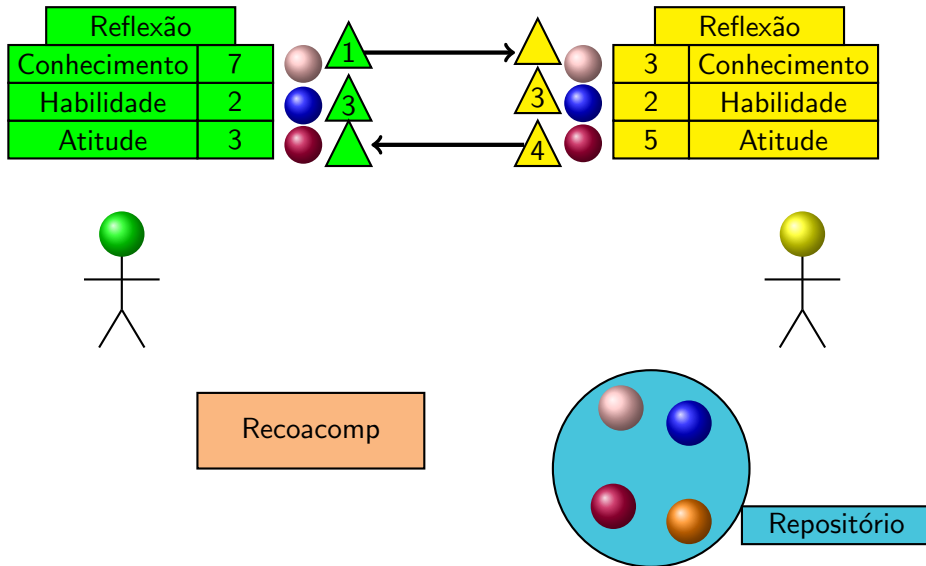


Repositório

Filtragem Colaborativa - Grau de Similaridade

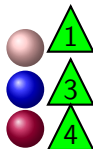
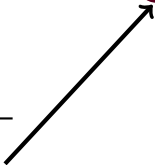
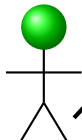


Filtragem Colaborativa - Recomendação

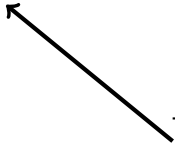
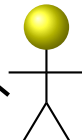
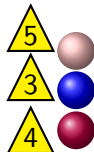


Filtragem Colaborativa - FeedBack dos Usuários

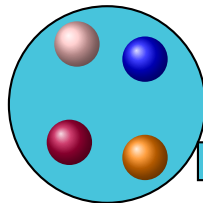
Reflexão	
Conhecimento	7
Habilidade	2
Atitude	3



Reflexão	
3	Conhecimento
2	Habilidade
5	Atitude



Recoacomp



Repositório

Model View Controller - MVC

JavaServer Faces - JSF

É um framework MVC baseado em Java para a construção de interfaces de usuário baseadas em componentes para aplicações web. Possui um modelo de programação dirigido a eventos, abstraindo os detalhes da manipulação dos eventos e organização dos componentes, permitindo que o programador se concentre na lógica da aplicação.

MVC Pull - Action Based

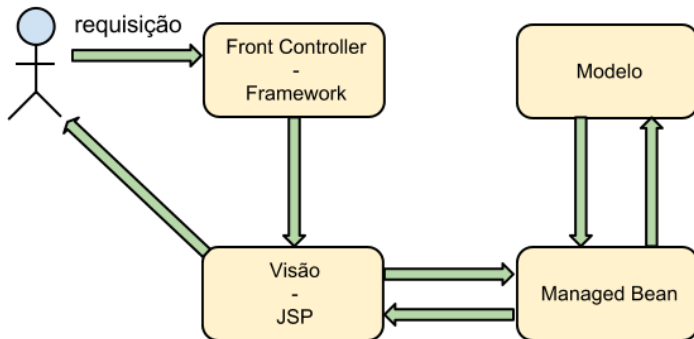


Figura: Modelo MVC Pull

View

Facelets

Facelets é uma linguagem de declaração página, que é usado para construir JavaServer Faces exibições usando modelos de estilo HTML e para a construção de árvores de componentes.

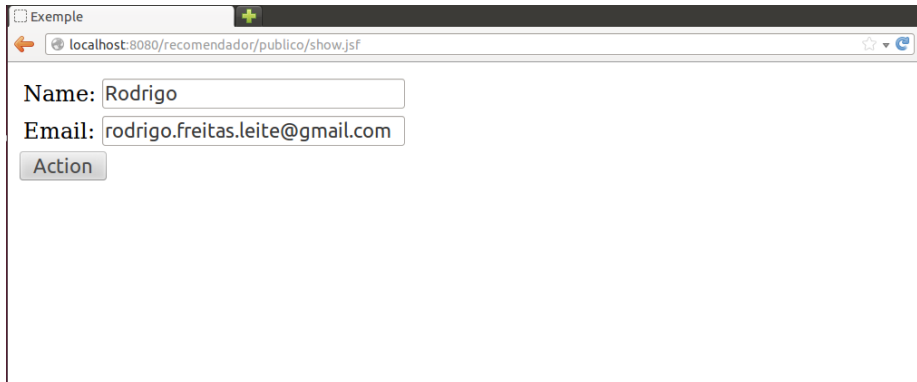
Facelets - Página

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:ui="http://java.sun.com/jsf/facelets">

  <h:head>
    <title>Exemple</title>
  </h:head>

  <h:body>
    <h:form>
      <h:panelGrid columns="2">
        <h:outputText value="Nome: " />
        <h:inputText value="#{user.name}" />
        <h:outputText value="Email: " />
        <h:inputText value="#{user.email}" />
      </h:panelGrid>
      <h:commandButton value="action" action="#{user.action()}" />
    </h:form>
  </h:body>
</html>
```


Página



A screenshot of a web browser window. The address bar shows the URL `localhost:8080/recomendador/publico/show.jsf`. The page content includes a form with two text input fields and one button. The first field is labeled "Name:" and contains the text "Rodrigo". The second field is labeled "Email:" and contains the text "rodrigo.freitas.leite@gmail.com". Below these fields is a button labeled "Action".

Example

localhost:8080/recomendador/publico/show.jsf

Name:

Email:

Figura: Página Gerada

Criando Componente

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:composite="http://java.sun.com/jsf/composite"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:ui="http://java.sun.com/jsf/facelets">

<body>

<composite:interface>

<composite:attribute name="typeAction" />

</composite:interface>
<composite:implementation>

<h:panelGrid columns="2" >
<h:outputText value="Name: " />
<h:inputText value="#{user.name}" />
<h:outputText value="Email: " />
<h:inputText value="#{user.email}" />
</h:panelGrid>
<h:commandButton value="#{cc.attrs.typeAction}" action="#{user.action()}" />

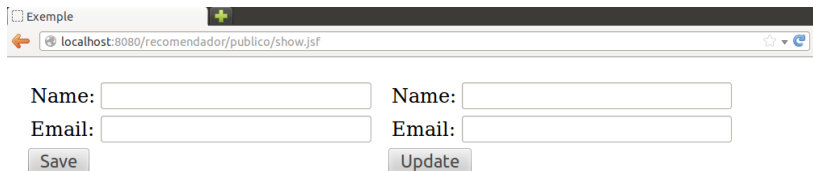
</composite:implementation>

</body>
</html>
```

Utilizando Componente

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:util="http://java.sun.com/jsf/composite/components/util">
<h:head>
<title>Exemple</title>
</h:head>
<h:body>
<h:form >
<util:componente typeAction="Save" />
<util:componente typeAction="Update" />
</h:form>
</h:body>
</html>
```

Página com Componente



The screenshot shows a web browser window with the title 'Exemple' and a tab icon. The address bar displays 'localhost:8080/recomendador/publico/show.jsf'. The main content area contains two identical form components side-by-side. Each component has a 'Name:' label followed by a text input field, an 'Email:' label followed by a text input field, and a button below the email field. The left component has a 'Save' button, and the right component has an 'Update' button.

Figura: Página com Componente

Diagrama Entidade-Relacionamento

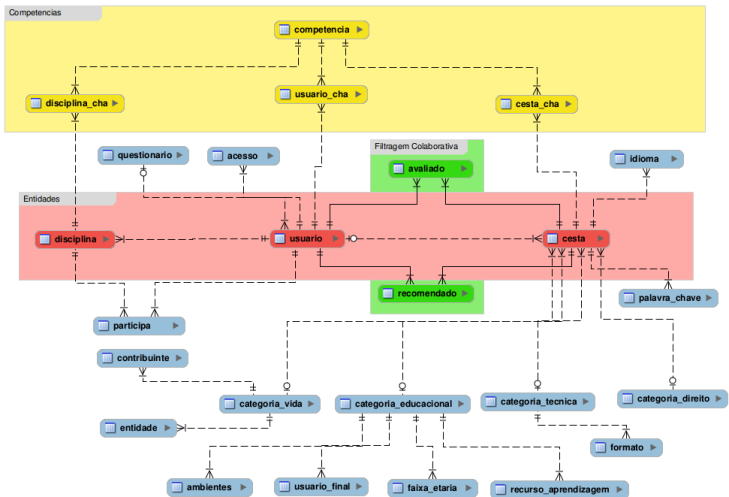


Figura: Diagrama de Entidade Relacionamento

Mapeamento Objeto Relacional

Java Persistence API

É uma especificação que descreve o gerenciamento de dados relacionais em aplicações. Tal gerenciamento é realizado de maneira automatizada e transparente para tabelas em um banco de dados relacional, usando metadados que descrevem o mapeamento entre objetos e o banco de dados.

Mapeamento Objeto Relacional

```
1 @Entity
2 public class usuario {
3     @Id
4     @Column
5     private int id;
6
7     @Column
8     private String nome;
9
10    @Column
11    private String senha;
12
13    @Column
14    private String email;
15
16    @OneToMany
17    private List<UsuarioCha> cha;
18 }
```

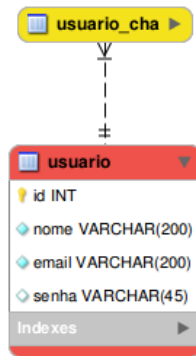


Figura: Tabela Usuario

Perguntas?

Recomendador de Objetos de Aprendizagem Baseado em Competências Recoacomp

Rodrigo Freitas Leite

UFRGS-NUTED

4 de Outubro de 2013

rodrigo.freitas.leite@gmail.com