



Disciplina: Sistemas Distribuídos

Professora: Ana Cristina Barreiras Kochem Vendramin

Avaliação (valor 2,5)
Arquitetura Cliente-Servidor, Modelo *Publisher/Subscriber* (Eventos e Notificações), Java RMI

Agência de Empregos

Desenvolver uma aplicação cliente-servidor que gerencie uma agência de empregos.

Requisitos da aplicação:

- Utilizar a *middleware* **Java RMI** (*Remote Invocation Method*) ou **Pyro** (*Python Remote Objects*) para prover a comunicação entre os clientes e o servidor da aplicação de agência de empregos;
- Métodos disponíveis no servidor (valor 1,8):
 - **Consulta (valor 0,6):**
 - **Consulta de vagas de emprego**, indicando filtros como área de interesse e salário mínimo pretendido (valor 0,3).
 - **Consulta de currículos**, indicando a área de interesse (valor 0,3).
 - **Cadastro (valor 0,6):**
 - Para clientes que procuram uma vaga de emprego - **Cadastro e alteração de currículo** (fornecendo nome, contato, área de interesse, carga horária e salário pretendido) (valor 0,3).
 - Para clientes que oferecem uma vaga de emprego - **Cadastro e alteração de vaga de emprego** (fornecendo nome da empresa, contato, a área da vaga, carga horária e salário) (valor 0,3).
 - **Registro de interesse em eventos (valor 0,6):**
 - O cliente interessado em uma vaga de emprego pode registrar interesse em receber notificações do servidor quando uma nova vaga na área de seu interesse estiver disponível. Para isso, o cliente (assinante) enviará a sua referência de objeto remoto ao servidor e sua área de interesse (valor 0,3).
 - O cliente interessado em profissionais pode registrar interesse em receber notificações do servidor quando um novo profissional do seu interesse estiver disponível. Para isso, o cliente (assinante) enviará a sua referência de objeto remoto ao servidor e área da oferta de emprego (valor 0,3).

- Existem dois tipos de clientes na aplicação: o cliente interessado em uma vaga de emprego e o cliente que oferece uma vaga de emprego. Para cada cliente deve-se criar, no mínimo: um projeto, um pacote, uma classe principal, uma interface e uma classe servente. **Cada cliente** tem um método para o **recebimento de notificações de eventos do servidor (valor 0,7)**:
 - O cliente interessado em uma vaga de emprego receberá uma notificação de evento do servidor, via **chamada de método**, quando surgir uma vaga na sua área de interesse (valor 0,35);
 - O cliente que oferece uma vaga de emprego receberá uma notificação de evento do servidor, via **chamada de método**, quando surgir um currículo que atenda alguma de suas vagas de emprego (valor 0,35);

Observações:

- Desenvolva uma interface com recursos de interação apropriados;
- Gerar a documentação de todos os métodos de sua aplicação (obs.: será descontado 0,5 pontos na ausência da documentação);
- Equipe: dois programadores.