



**Prof.<sup>a</sup> Ana Cristina Barreiras Kochem Vendramin**

Universidade Tecnológica Federal do Paraná (UTFPR), Câmpus Curitiba  
Departamento Acadêmico de Informática (DAINF)

**Avaliação (valor 3,0)**  
**Arquitetura Cliente-Servidor**  
**Modelo *Publisher/Subscriber* (Eventos e Notificações)**

**Aplicação distribuída para viagens de carona**

- Utilize a *middleware* **Java RMI** (*Remote Invocation Method*) ou **PyRO** (*Python Remote Objects*) para prover a comunicação entre os processos.
- Execute um processo servidor e ao menos dois processos clientes. Esses processos podem ser executados na mesma máquina.
- Existem dois tipos de clientes na aplicação: o cliente interessado em uma carona e o cliente que oferece uma carona.
- Haverá apenas um serviço de nomes na máquina. O processo servidor criará esse serviço de nomes e registrará a referência de sua aplicação nele. Cada processo cliente conhece o nome da aplicação do servidor (exemplo: "Carona") para poder obter junto ao serviço de nomes a referência do objeto remoto através do seu nome (método *lookup* no Java RMI ou *resolve* no PyRO). Uma alternativa é utilizar o método *list* (no Java RMI ou PyRO) ou *flatlist* (no PyRO) do serviço de nomes para obter os nomes das aplicações que estão cadastradas no serviço de nomes e, posteriormente, chamar o *lookup* (Java RMI) ou *resolve* (PyRO) para obter a respectiva referência de objeto remoto.
- Métodos disponíveis no **servidor (valor 2,0)**:
  - **Cadastro usuário (valor 0,2)**:
    - Clientes devem informar seu nome, telefone e chave pública.
  - **Consulta de caronas (valor 0,3)**:
    - **Consulta de caronas.** Clientes devem informar a origem, destino e a data da viagem desejada.
  - **Registro de interesse em eventos (valor 1,1)**:
    - O cliente interessado em uma carona pode atuar como *subscriber*, registrando interesse em receber notificações do servidor quando uma nova carona que atenda seus critérios de viagem (evento de interesse) estiver disponível. Para isso, o cliente (assinante) enviará a sua referência de objeto remoto ao servidor juntamente com seu nome, contato, origem, destino e data da viagem. O servidor retornará um id único para esse registro (**valor 0,3**).
      - Essa mensagem de registro de interesse em carona deve conter uma assinatura digital. Para isso, o cliente

utilizará a sua chave privada. O servidor receberá a mensagem e validará a assinatura utilizando a chave pública correspondente (**valor 0,25**).

- O cliente interessado em passageiros para sua viagem pode atuar como *subscriber*, registrando interesse em receber notificações do servidor quando um novo passageiro que atenda seus critérios de viagem (evento de interesse) estiver disponível. Para isso, o cliente (assinante) enviará a sua referência de objeto remoto ao servidor juntamente com seu nome, contato, origem, destino, data da viagem e número de passageiros. O servidor retornará um id único para esse registro (**valor 0,3**).
  - Essa mensagem de registro de interesse em passageiros para viagem deve conter uma assinatura digital. Para isso, o cliente utilizará a sua chave privada. O servidor receberá a mensagem e validará a assinatura utilizando a chave pública correspondente (**valor 0,25**).
- **Cancelamento de um registro de interesse (valor 0,4):**
  - O cliente interessado em uma carona pode a qualquer momento cancelar o interesse em receber notificações sobre novas caronas de seu interesse. Para isso será informado o id do registro que se deseja cancelar (**valor 0,2**).
  - O cliente interessado em passageiros pode a qualquer momento cancelar o interesse em receber notificações sobre novos passageiros de seu interesse. Para isso será informado o id do registro que se deseja cancelar (**valor 0,2**).
- Cada **cliente** tem um método para o **recebimento de notificações de eventos do servidor (valor 1,0)**:
  - O cliente interessado em uma carona receberá uma notificação de evento do servidor, via **chamada de método**, quando surgir uma carona que atenda seus critérios. Essa mensagem conterá informações sobre o motorista que está oferecendo a carona (**valor 0,5**);
  - O cliente que oferece uma carona receberá uma notificação de evento do servidor, via **chamada de método**, quando surgir um passageiro interessado na sua carona. Essa mensagem conterá informações sobre o passageiro que está interessado na sua carona (**valor 0,5**);

Observações:

- Desenvolva uma interface com recursos de interação apropriados;
- Adicione comentários no código explicando os métodos de sua aplicação;
- É obrigatória a defesa da aplicação para obter a nota.
- O desenvolvimento da aplicação pode ser individual ou em dupla. Porém, a defesa da aplicação é individual.