

Trabalho de Implementação de Objetos Distribuídos

Desenvolver um sistema usando o middleware CORBA com as seguintes especificações:

- 1) **Servidor de Recurso**, tem uma coleção de registros (cada registro contém chave e valor). A chave nunca se repete na lista. Deve conter uma lista própria de pelo menos 20 registros que estão disponíveis para serem acessados.
 - a. **Construtor:** recebe o endereço e nome do DNS. Recebe o nome desse objeto de recurso, e a porta.
 - b. **Método remoto getValor:** recebe como parâmetro um string **chave** e retorna o **valor** do registro. Deve pesquisar na lista por um registro cuja chave seja a mesma, e deve retornar o valor do registro.
 - c. Cada objeto de recursos deve ter um nome diferente.
 - d. Toda a vez que um objeto de recursos for iniciado, ele deve se cadastrar no DNS com seu nome, ip e porta, usando o método **DNS.add**.
 - e. Quando o objeto de recursos for finalizado, ele deve se remover do dns usando **DNS.unbind**.
 - f. Exemplo de lista de registros:

Chave	Valor
Processador	Pentium
Memória	Kingston
Hard Drive	Samsung
 - g. O objeto ServidorRecurso deve registrar na tela o resultado das chamadas de seus métodos remotos para conferência.
- 2) **Objeto Cliente**. É um objeto interativo com o usuário que tem a função de consultar registros e mostrar o valor dos registros. O usuário digita uma chave e é feita a consulta do valor no objeto de recursos e mostrado o valor na tela do usuário. Pode ser uma aplicação gráfica ou console.
 - a. **Construtor:** recebe o endereço e nome do DNS. Recebe o nome desse objeto de recurso, e a porta.
 - b. **Construtor:** recebe o seu nome, ip e porta.
 - c. **A consulta de chaves funciona da seguinte forma:** o usuário digita uma chave na tela e manda consultar. O cliente chama o método **DNS.getServidoresRecurso** para consultar os nomes dos objetos de recurso disponíveis, retornando um vetor com o nome de todos os objetos de recurso. Em seguida o cliente chama o método **DNS.bind(nome)** para obter o ip e porta de cada servidor de recursos. Para cada servidor de recurso, o cliente chama o método **ServidorRecurso.getValor(chave)** para obter o valor da chave. Se encontrou a chave nesse servidor, mostra na tela o valor da chave e termina a busca.
 - d. Quando um novo cliente for iniciado, ele deve se registrar no DNS.
 - e. O Objeto Cliente deve registrar na tela todas as chamadas de métodos remotos que ele fizer.
 - f. Quando o cliente for finalizado, ele deve se remover do DNS usando o método **DNS.unbind**.
- 3) **Objeto de Resolução de Nomes (DNS)**. Ele guarda uma lista de nomes e IPs relacionados aos mesmos, além do tipo do objeto de recurso (e a porta).

- a. **Método bind:** recebe como parametro um nome e retorna o ip e porta do objeto remoto.
 - b. **Método unbind:** recebe como parametro um nome e remove o mapeamento nome, ip, porta.
 - c. **Método add:** recebe um nome, ip e porta e cria o mapeamento. Os mapeamentos serão uma lista de objetos.
 - d. **Método getServidoresRecurso:** retorna um vetor com os nomes dos servidores de recursos do sistema.
 - e. Todos os objetos distribuídos podem ter fixo o endereço e porta do DNS.
 - f. O DNS deve registrar na tela todas as chamadas de seus métodos remotos para conferência.
- 4) **Objeto com método main:**
- a. O endereço do DNS é uma variável global estática.
 - b. Deve criar um único DNS.
 - c. Deve criar três objetos de recursos com lista de chaves diferentes entre si.
 - d. Deve criar um cliente.