

Sistemas Distribuídos 2024/2025

Ficha Prática Nº 10 – REST Web Services (cont.)

1. Implementar um **servidor** e um **cliente REST** para manter informações de preços e variação no preço de ações. O programa servidor deve utilizar uma classe básica chamada Acao, conforme mostrado abaixo:

```
public class Acao{
    private String sigla;    // exemplo: EDPR
    private String nome;    // exemplo: EDP Renováveis, SA
    private float  valor_abertura; // exemplo: 8.39 €
    private float  valor_atual;    // exemplo: 8.30 €
    public Acao(String sigla, String nome, float valor_abertura){...}
    public float getVariacao(){...} // exemplo: -1.07%
    // gets and sets, se necessários
    ...
};
```

O servidor deve disponibilizar a seguinte funcionalidade:

- Obter os dados de todas as ações com o método GET.
- Obter os dados de uma ação, pela sigla, com o método GET.
- Obter os dados da ação com maior variação positiva, com o método GET.
- Adicionar uma ação, com o método POST (dados do construtor).
- Atualizar o valor atual de uma ação, com o método PUT.
- Remover uma ação, com o método DELETE.

Observações:

- a) A sigla da ação deve ser utilizada para definir a URI do recurso. Não deve ser possível adicionar duas ações com a mesma sigla ou com o mesmo nome. O nome e sigla da ação não podem estar em branco.
- b) O valor de uma ação não pode ser negativo ou zero e deve ter duas casas decimais. A variação de uma ação deve ser expressa em percentagem, pode ser positiva ou negativa, e deve ter duas casas decimais.
- c) Nas consultas por ação, devem ser apresentadas ao utilizador as seguintes informações: sigla, nome, valor atual e variação. As consultas de todas as ações devem ser apresentadas em forma de tabela.

- d) As respostas do servidor devem retornar o código HTTP apropriado, mesmo em caso de erro. O programa cliente deve converter o código de resposta HTTP numa mensagem apropriada para apresentação ao utilizador.
- e) Na troca de dados de ações entre o cliente e o servidor deve ser utilizado o formato XML ou JSON. A apresentação de dados ao utilizador deve ser independente do formato de dados utilizado na comunicação.
- f) O sistema não deve estar sujeito a “race conditions”. Explicar os mecanismos de exclusão mútua utilizados (se forem necessários). Apresentar os testes realizados e resultados obtidos (com screenshots).

Submeter dois projetos Netbeans (cliente e servidor) e o relatório em pdf. O servidor deve estar configurado para executar sobre o GlassFish Server 7.

O trabalho é **individual**. Aplicam-se as mesmas regras sobre **cópia/fraude** descritas nas fichas práticas 1 e 2.