

## Resumo dos Exemplos

### ✓ Exemplo 1: Envio de dados com método GET

URL:

http://localhost:8080/arquivos/arquivo.jsp?txtNome=Carlos&txtIdade=10&txtSalario=1000

Código (**arquivo.jsp**):

```
<%@ page import="java.io.*" %>
<%
    String nome = request.getParameter("txtNome");
    int idade = Integer.parseInt(request.getParameter("txtIdade"));
    double salario = Double.parseDouble(request.getParameter("txtSalario"));

    try (FileWriter f = new FileWriter("c:/arquivo.txt", true)) {
        f.write(nome + ";" + idade + ";" + salario + "\r\n");
    } catch (Exception e) {
        <h1>OCORREU UM ERRO AO TENTAR GRAVAR O ARQUIVO</h1>
    }
%>
```



### Observações:

- O método **GET** envia os dados pela URL.
- Não é necessário formulário: você pode digitar a URL diretamente no navegador.
- Dados são salvos em **c:/arquivo.txt**.

---

### ✓ Exemplo 2: Leitura de dados de um arquivo

Código (**abreArquivo.jsp**):

```
<%@ page import="java.io.*" %>
```

```

<%
    try (BufferedReader br = new BufferedReader(new FileReader("c:/arquivo.txt"))) {
        String linha;
        while ((linha = br.readLine()) != null) {
            String[] dados = linha.split(";");
%>
            <b>Nome:</b> <%= dados[0] %><br>
            <b>Idade:</b> <%= dados[1] %><br>
            <b>Salário:</b> <%= dados[2] %><br><hr>
<%
        }
    } catch (Exception e) {
%>
        <h1>ERRO AO LER ARQUIVO!</h1>
<%
    }
%>

```



### Observações:

- Não precisa de parâmetros.
- Lê o arquivo `c:/arquivo.txt` e exibe os dados no navegador.



### Exemplo 3: Envio de dados com método POST (via formulário)

HTML (`formulario.html` ou embutido em `index.jsp`):

```

<form method="post" action="arquivo.jsp">
    Nome: <input name="txtNome" type="text"><br>
    Idade: <input name="txtIdade" type="text"><br>
    Salário: <input name="txtSalario" type="text"><br>
    <input type="submit" value="Enviar">
</form>

```

### Comportamento:

- Envia os dados para o mesmo `arquivo.jsp` do exemplo 1.
- Usa o método POST (os dados não ficam visíveis na URL).
- Mais seguro para formulários do que o método GET.



## Dicas importantes:



### Práticas desatualizadas:

- Uso de **scriptlets** (`<% ... %>`) está **obsoleto** em projetos modernos.
- Gravando arquivos diretamente no `*C:*` é perigoso e inseguro (problemas de permissão e segurança).



### Recomendado atualmente:

- Usar **Servlets + JSP com JSTL e EL**.
- Substituir `FileWriter` por acesso via banco de dados (ex: JDBC ou JPA).
- Para aplicações modernas, prefira **Spring Boot** com Thymeleaf, REST API, etc.