

Chegou a hora de você pôr em prática o que foi visto na aula. Para isso, siga os passos listados abaixo.

1) No pacote `br.com.alura.gerenciador.servlet`, crie o filtro `MonitoramentoFilter`, implementando a interface `Filter`. Dentro do método `doFilter`, calcule o tempo de execução da ação, então, guarde o tempo do sistema em uma variável, execute a ação e após isso, guarde o tempo do sistema em uma nova variável. Por fim, faça a subtração da segunda variável com a primeira, calculando assim o tempo de execução da ação e imprima o resultado.

2) Também no pacote `br.com.alura.gerenciador.servlet`, crie o filtro `AutorizacaoFilter`, implementando a interface `Filter`. Dentro do método `doFilter`, faça a verificação e autorização do usuário logado, semelhante à que é feita na classe `UnicaEntradaServlet`, e continue com o fluxo dos filtros. Preste atenção ao `ServletRequest` e ao `ServletResponse`, pois será necessário fazer um `cast` para `HttpServletRequest` e `HttpServletResponse`, respectivamente. Você também precisará ler o parâmetro `acao` da requisição.

3) Em `UnicaEntradaServlet`, remova o mapeamento para `/entrada` e a verificação e autorização do usuário logado:

```
// além do mapeamento, remova as seguintes linhas

HttpSession sessao = request.getSession();
boolean usuarioNaoEstaLogado = (sessao.getAttribute("usuarioLogado") == null);
boolean ehUmaAcaoProtegida = !(paramAcao.equals("Login") || paramAcao.equals("logout"));

if(ehUmaAcaoProtegida && usuarioNaoEstaLogado) {
    response.sendRedirect("entrada?acao=LoginForm");
    return;
}
```

4) Crie o filtro `ControladorFilter`, no pacote `br.com.alura.gerenciador.servlet`, implementando a interface `Filter`. Copie o código do método `service`, da classe `UnicaEntradaServlet`, e cole dentro do método `doFilter` da nova classe, não esquecendo de, antes, fazer o `cast` de `ServletRequest` e `ServletResponse` para `HttpServletRequest` e `HttpServletResponse`, respectivamente

5) Por fim, faça o mapeamento dos filtros no **web.xml**:

```
<filter>
    <filter-name>MF</filter-name>
    <filter-class>br.com.alura.gerenciador.servlet.MonitoramentoFilter</filter-class>
</filter>

<filter-mapping>
    <filter-name>MF</filter-name>
    <url-pattern>/entrada</url-pattern>
</filter-mapping>

<filter>
    <filter-name>AF</filter-name>
    <filter-class>br.com.alura.gerenciador.servlet.AutorizacaoFilter</filter-class>
</filter>

<filter-mapping>
    <filter-name>AF</filter-name>
    <url-pattern>/entrada</url-pattern>
</filter-mapping>

<filter>
    <filter-name>CF</filter-name>
    <filter-class>br.com.alura.gerenciador.servlet.ControladorFilter</filter-class>
</filter>

<filter-mapping>
    <filter-name>CF</filter-name>
    <url-pattern>/entrada</url-pattern>
</filter-mapping>
```

## Opinião do instrutor

A classe `MonitoramentoFilter` ficará assim:

```
public class MonitoramentoFilter implements Filter {

    @Override
    public void doFilter(ServletRequest request, ServletResponse response
        throws IOException, ServletException {

        System.out.println("MonitoramentoFilter");

        long antes = System.currentTimeMillis();

        String acao = request.getParameter("acao");

        chain.doFilter(request, response);

        long depois = System.currentTimeMillis();
        System.out.println("Tempo de execução da acao " + acao + " -> " + (depois - antes));
    }
}
```

A classe `AutorizacaoFilter` ficará assim:

```
public class AutorizacaoFilter implements Filter {

    public void doFilter(ServletRequest servletRequest, ServletResponse s
        throws IOException, ServletException {

        System.out.println("AutorizacaoFilter");

        HttpServletRequest request = (HttpServletRequest) servletRequest;
        HttpServletResponse response = (HttpServletResponse) servletRespo

        String paramAcao = request.getParameter("acao");

        HttpSession sessao = request.getSession();
        boolean usuarioNaoEstaLogado = (sessao.getAttribute("usuarioLogado") == null);
        boolean ehUmaAcaoProtegida = !(paramAcao.equals("Login") || paramAcao.equals("logout"));

        if(ehUmaAcaoProtegida && usuarioNaoEstaLogado) {
            response.sendRedirect("entrada?acao=LoginForm");
            return;
        }

        chain.doFilter(request, response);
    }
}
```

E a classe `ControladorFilter` ficará assim:

```
public class ControladorFilter implements Filter {

    public void doFilter(ServletRequest servletRequest, ServletResponse s
        throws IOException, ServletException {

        System.out.println("ControladorFilter");
        HttpServletRequest request = (HttpServletRequest) servletRequest;
        HttpServletResponse response = (HttpServletResponse) servletRespo

        String paramAcao = request.getParameter("acao");
        String nomeDaClasse = "br.com.alura.gerenciador.acao." + paramAcao;

        String nome;
        try {
            Class classe = Class.forName(nomeDaClasse);//carrega a classe
            Acao acao = (Acao) classe.newInstance();
            nome = acao.executa(request,response);
        } catch (ClassNotFoundException | InstantiationException | IllegalAccessException e) {
            throw new ServletException(e);
        }
        String[] tipoEEndereco = nome.split(":");
        if(tipoEEndereco[0].equals("forward")) {
            RequestDispatcher rd = request.getRequestDispatcher("WEB-INF/" + tipoEEndereco[1]);
            rd.forward(request, response);
        } else {
            response.sendRedirect(tipoEEndereco[1]);
        }
    }
}
```

Continue com os seus estudos, e se houver dúvidas, não hesite em recorrer ao nosso fórum!