

Servidor

```
import java.net.*;
import java.io.*;

public class ChatHttpServer {

    public static void main( String[] args) throws IOException {
        int serverPort = 8080;
        if( args.length > 1) {
            System.err.println( "Use: java ChatHttpServer [porto]");
            System.exit(0);
        } else if( args.length == 1)
            serverPort = Integer.parseInt( args[0]);
        new ChatHttpServer( serverPort).doIt();
    }

    protected int serverPort;
    protected ChatUtilities chat;

    protected ChatHttpServer( int serverPort) throws IOException {
        this.serverPort = serverPort;
        this.chat = new ChatUtilities( serverPort);
    }

    // ===== funções auxiliares =====
    /*
    * Copia o conteudo de um canal de entrada para um canal de saída...
    */

    protected void dumpStream( InputStream in, OutputStream out) throws IOException {
        ...
    }

    // ===== funções principais =====
    void processRequest( Socket socket) {
        //TODO: completar com o processamento do pedido
    }

    void doIt() {
        //TODO: Completar com codigo para esperar as conexoes e que
        // irá chamar processRequest() para tratar o pedido...
    }
}
```

HTTPUtilities

```
import java.io.* ;
import java.net.* ;
import java.util.* ;

public class HTTPUtilities {

    /**
     * Consome uma linha de texto terminada por CR do InputStream dado
     * e devolve-a numa String.
     */
    public static String readLine( InputStream is ) throws IOException {
        ...
    }

    /** A partir da linha de pedido HTTP devolve um array com o metodo invocado
     * na posicao 0, o URL na posicao 1, e a versao na posicao 2 Por ex., para
     * "GET http://www.unl.pt/index.html HTTP/1.0", devolve [0]->"GET";
     * [1]->"http://www.unl.pt/index.html"; [2]->"HTTP/1.0" Caso request nao
     * seja um pedido HTTP valido, devolve null.
     */
    public static String[] parseHttpRequest( String request) {
        ...
    }

    /** A partir de um header HTTP, devolve um array com o nome do header na
     * posicao 0 e o valor indicado na posicao 1 Por ex., para "Connection:
     * Keep-alive", devolve [0]->"Connection"; [1]->"Keep-alive" */
    public static String[] parseHTTPHeader( String header) {
        ...
    }

    /**
     * A partir de uma string com o conteúdo de um form submetido no formato
     * application/x-www-form-urlencoded, devolve um objecto do tipo Properties,
     * associando a cada elemento do form o seu valor
     */
    public static Properties parseHttpPostContents( String contents) throws IOException {
        ...
    }
}
```

ChatUtilities

```
import java.io.* ;
import java.net.* ;
import java.util.* ;

public class ChatUtilities {
    ...
    public ChatUtilities( int serverPort) {
        ...
    }
    /** Adiciona a mensagem message, enviada pelo utilicador nickName, ano chat
     * room chatName
     */
    public void addChatMessage( String chatName, String nickName, String message) {
        ...
    }

    /**
     * Devolve um input stream com o conteudo HTML que pode ser usado como
     * resposta em caso de erro.
     */
    public InputStream getErrorPageInputStream() {
        ...
    }

    /**
     * Devolve um input stream com o conteudo HTML da pagina do form
     */
    public InputStream getFormPageInputStream( String chatName, String nickName) throws IOException {
        ...
    }

    /**
     * Devolve um input stream com o conteudo HTML da pagina principal
     */
    public InputStream getMainPageInputStream( String chatName) throws IOException {
        ...
    }
}
```