

Manual de uso do HighFrame

Artêmio Oliveira de Andrade Junior
Felipe Oliveira Carvalho
Saulo Eduardo Galilleo Souza dos Santos
Tarcísio da Rocha
Victor Machado Pasquialino

São Cristóvão – SE

2014

Sumário

Lista de Figuras	ii
1 Manual de Uso	1
1.1 Requisitos para instalação	1
1.2 Download e Configuração	1
1.3 Iniciando os serviços	1
1.4 Utilização	2
1.5 Comanche WebServer	8

Lista de Figuras

1.1	Tela inicial do HighFrame Designer	3
1.2	Menu File	3
1.3	Submenu New	4
1.4	Submenu Open	4
1.5	Caixa para procura de arquivos	4
1.6	Menu Edit	5
1.7	Tela de criação de uma Arquitetura	5
1.8	Elementos de um componente	6
1.9	Tela de criação de uma Subarquitetura	7
1.10	Bind entre componentes	7
1.11	Bind entre subarquiteturas	8
1.12	Arquitetura distribuída do Comanche WebServer montada no HighFrame Designer	9

Capítulo 1

Manual de Uso

Nesta seção será demonstrada a correta configuração inicial para o uso do HighFrame Designer e um tutorial para sua utilização.

1.1 Requisitos para instalação

Para utilizar o HighFrame é preciso ter instalado o JDK do Java SE versão 1.7.0 ou superior. Caso não possua o JDK no seu equipamento ou não esteja atualizado efetue o download do instalador do JDK no site da Oracle <http://www.oracle.com/technetwork/pt/java/javase/downloads/jdk7-downloads-1880260.html>. O HighFrame designer pode ser executado nas plataformas Linux e Windows. Necessário apontar a variável de ambiente JAVA_HOME para o diretório de instalação do JDK.

Recomenda-se o uso de Processador Dual Core ou superior com no mínimo de 2 GB de memória RAM, disco rígido de 100 GB e sistema operacional Windows 7 ou 8 ou Linux.

1.2 Download e Configuração

O download do HighFrame Designer, HighFrame Client e HighFrame Server podem ser obtidos no seguinte endereço: <https://github.com/saulogalilleo/HighFrame/wiki/HighFrame>

1.3 Iniciando os serviços

Primeiramente é necessário executar o HighFrame Server. Para isso, abra o console de comandos (prompt ou shell) e entre no diretório onde os arquivos foram descompactados.

Em seguida digite o seguinte comando: `java -jar HighFrameServer.jar`

O próximo módulo distribuído a ser executado é o HighFrame Client. Abra um novo console de comandos (prompt ou shell) e de dentro do diretório onde os arquivos do HighFrame Client foram descompactados execute o seguinte comando: `java -jar HighFrameServer.jar`

Agora iremos iniciar o HighFrame Designer, após descompactar é necessário realizar a configuração do arquivo *properties.xml* o código 1.1 apresenta este arquivo.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <properties>
3     <subArchPath value="valor1" />
4     <deployPath value="valor2" />
5     <componentServerAddress value="valor4" />
6     <componentsPath value="valor3" />
7 </properties>
```

Código Fonte 1.1: *Properties.xml*

Neste arquivo será preciso informar os valores referente aos seguintes parâmetros de configuração:

- subArchPath: Caminho do diretório para armazenamento das subarquitecturas;
- deployPath: Caminho onde serão gerados os arquivos para o *deployment*;
- componentServerAddress: Endereço do servidor remoto de onde o HighFrame Designer irá obter a lista de componentes genéricos.
- componentsPath: Caminho do diretório onde o HighFrame Designer realizará o download dos componentes genéricos no formato XML.

Para os parâmetros subArchPath e DeployPath podem ser criadas novas pastas vazias. Após a configuração do arquivo properties.xml abra uma nova janela do console (prompt ou shell) e digite o seguinte comando: `java -jar HighFrameDesigner.jar`

1.4 Utilização

Serão demonstradas nesta seção algumas funcionalidades da IDE e suas respectivas telas e os significados dos elementos nelas contidas.

Ao abrir o HighFrame Designer a primeira tela que será mostrada é a tela contida na figura 1.1.

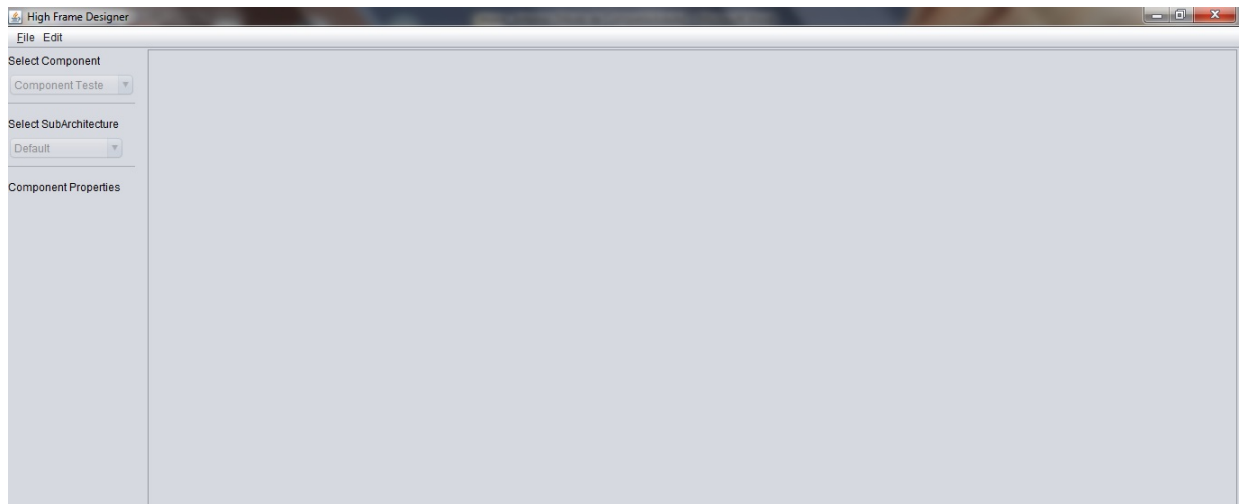


Figura 1.1: Tela inicial do *HighFrame Designer*

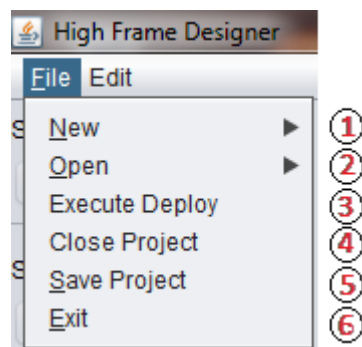


Figura 1.2: Menu *File*

- Menu **File** (Figura 1.2)

1. Novo projeto
2. Abrir projetos salvos
3. Executar o *deployment* do projeto atual
4. Fechar o projeto atual
5. Salvar o projeto atual
6. Sair do programa

- Opção **New** (Figura 1.3)

1. Criar nova arquitetura
2. Criar nova subarquitetura

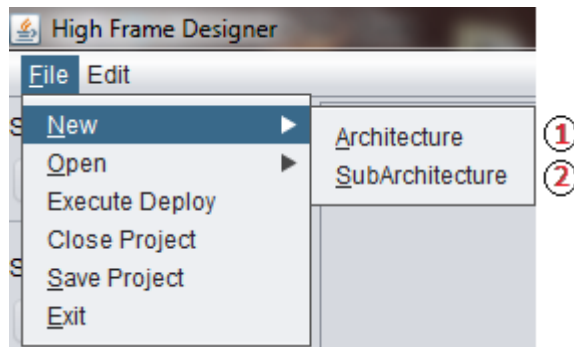


Figura 1.3: Submenu New

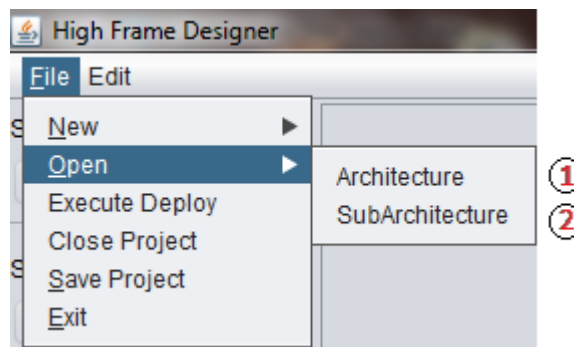


Figura 1.4: Submenu Open

- Opção **Open** (Figura 1.4)

1. Abrir arquitetura pré-existente
2. Abrir subarquitetura pré-existente

Ao escolher qualquer uma das opções acima o usuário deverá selecionar numa caixa igual a da figura 1.5 o arquivo de deseja abrir para trabalhar.

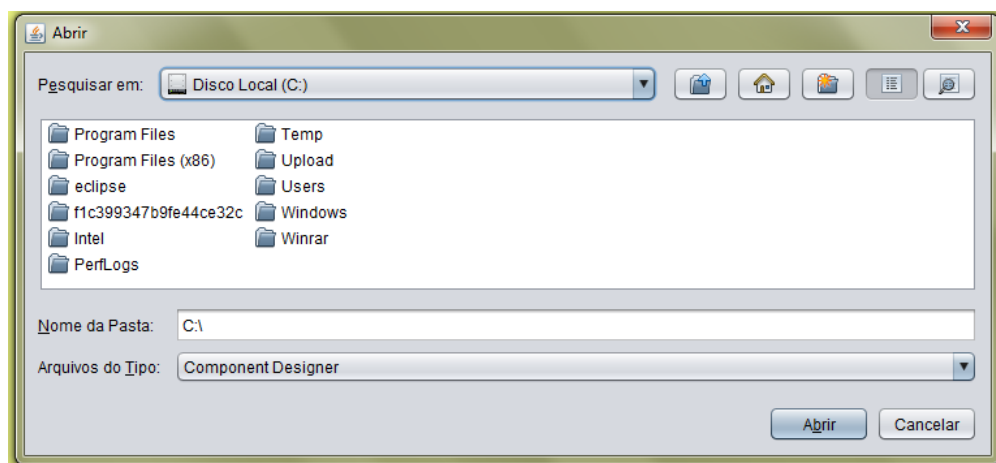


Figura 1.5: Caixa para procura de arquivos

Para executar o *deploy* de um projeto é preciso que o projeto desejado esteja aberto no HighFrame Designer e a opção *Execute Deploy* (item 3 da figura 1.2) seja selecionado. A partir dessa ação uma mensagem de sucesso ou erro será mostrada ao usuário a depender se o *deploy* foi executado corretamente.

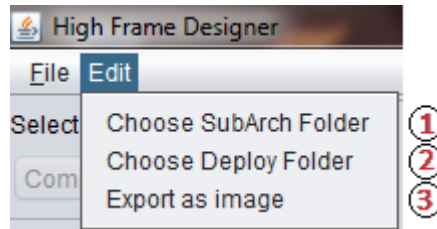


Figura 1.6: *Menu Edit*

- Menu **Edit** (Figura 1.6)

1. Selecionar pasta onde se encontram as subarquitecturas
2. Selecionar pasta onde será executado o *deploy*
3. Exportar modelo da arquitetura como imagem

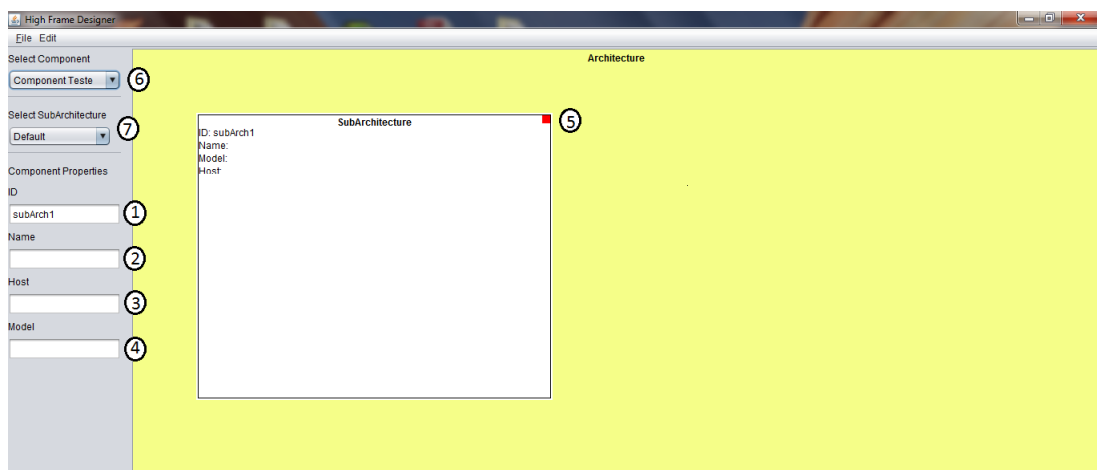


Figura 1.7: *Tela de criação de uma Arquitetura*

- **Elementos de uma Arquitetura** (Figura 1.7)

1. Identificação da subarquitectura
2. Nome da subarquitectura
3. Endereço da subarquitectura

4. Modelo de componente da subarquitetura
5. Remover subarquitetura
6. Lista de componentes pré-existentes
7. Lista de subarquitecturas pré-existentes

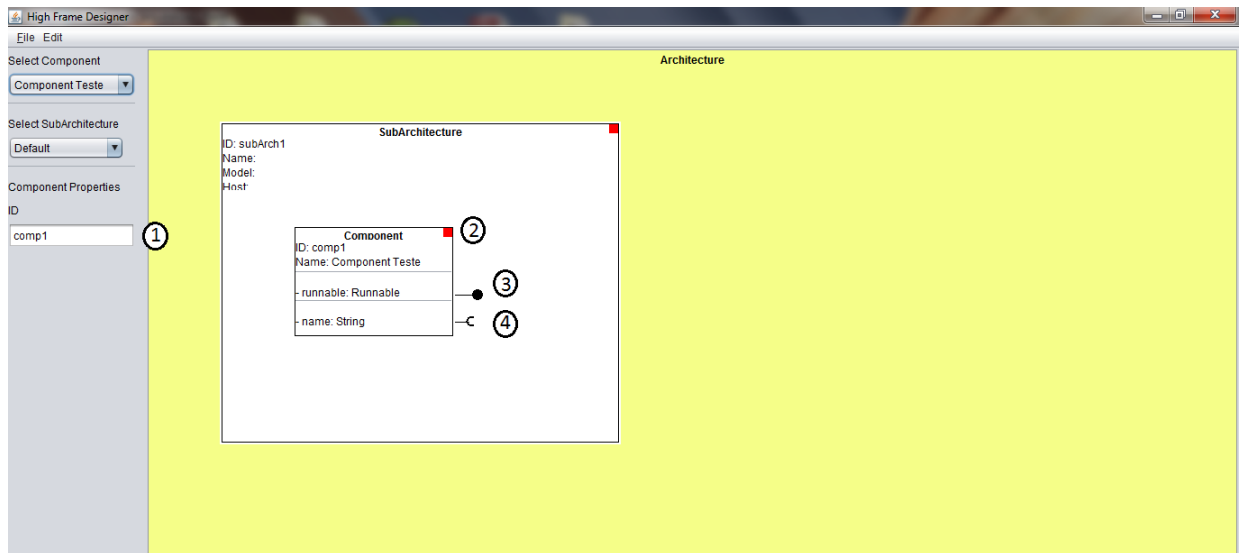


Figura 1.8: *Elementos de um componente*

- **Elementos de um Componente** (Figura 1.8)

1. Identificação do componente
2. Remover componente
3. Interface provida
4. Interface requerida

A criação de uma subarquitetura, como é mostrado na figura 1.9 segue os mesmos princípios da criação de uma arquitetura, com o diferencial de que apenas o menu de escolha de componentes fica habilitado.

Para adicionar uma ligação entre componentes é preciso clicar e manter pressionado o botão esquerdo do mouse na interface provida (item 3 da figura 1.8) do componente desejado e arrastar o cursor até a interface requerida (item 4 da figura 1.8) do outro componente desejado e soltar o botão esquerdo do mouse.

A ligação entre subarquitecturas é feita de forma análoga as ligações entre componentes como vemos na figura 1.11.

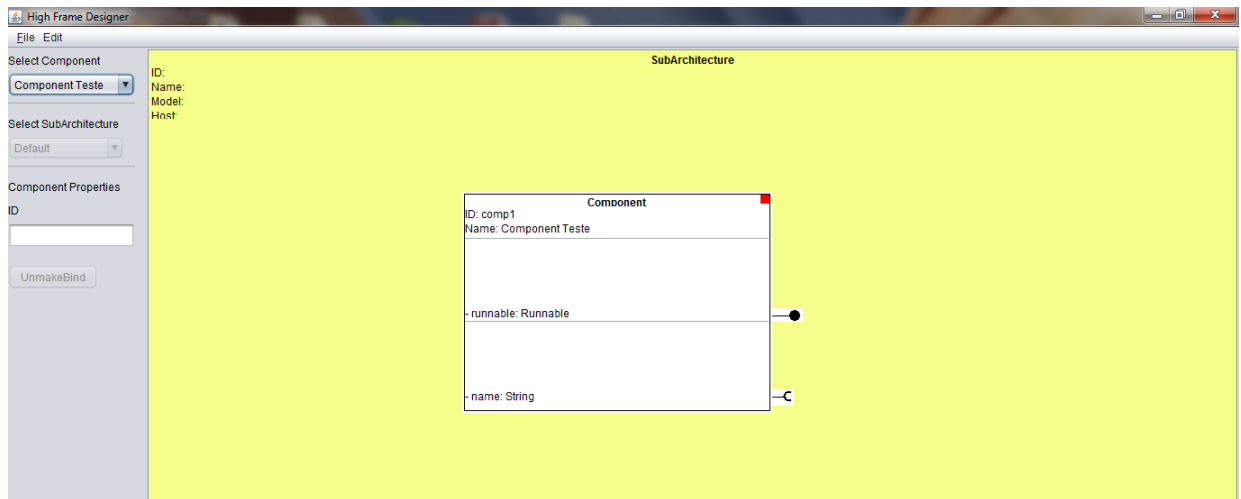


Figura 1.9: Tela de criação de uma Subarquitetura

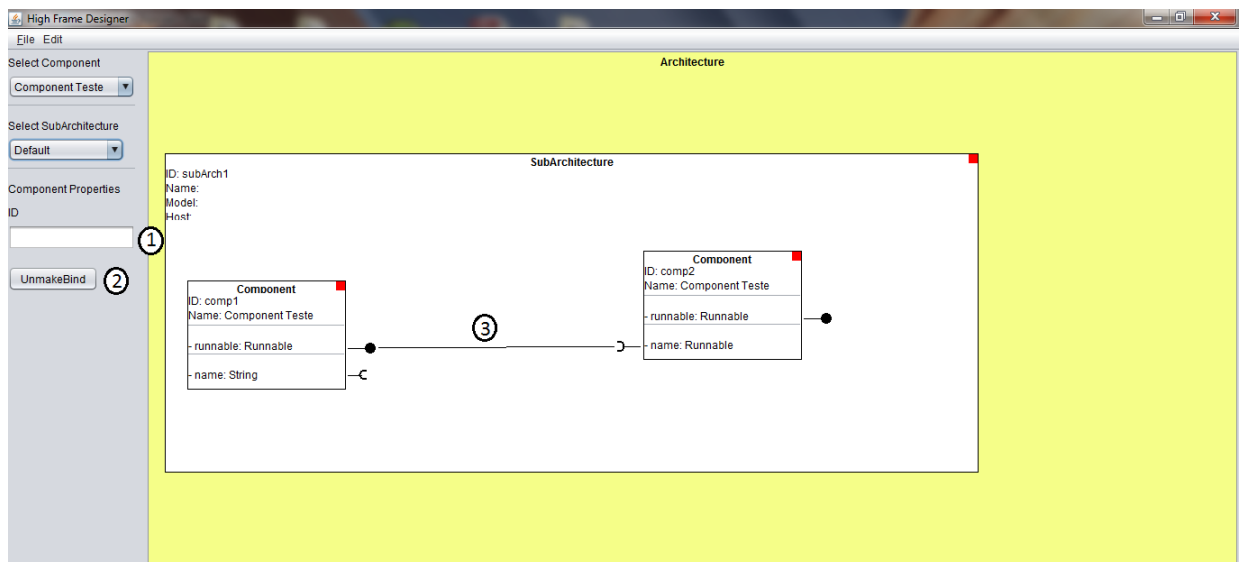


Figura 1.10: Bind entre componentes

- Elementos de um *bind* entre Componentes (Figura 1.10)

1. Identificação da interface
2. Desfazer *bind* para a interface selecionada
3. Representação gráfica do *bind*

Para desfazer uma ligação entre componentes ou subarquiteturas é preciso primeiramente selecionar a interface da ligação que se deseja remover, pode ser tanto a provida quanto a requerida, e depois clicar na opção *UnmakeBind* (item 2 da figura 1.10).

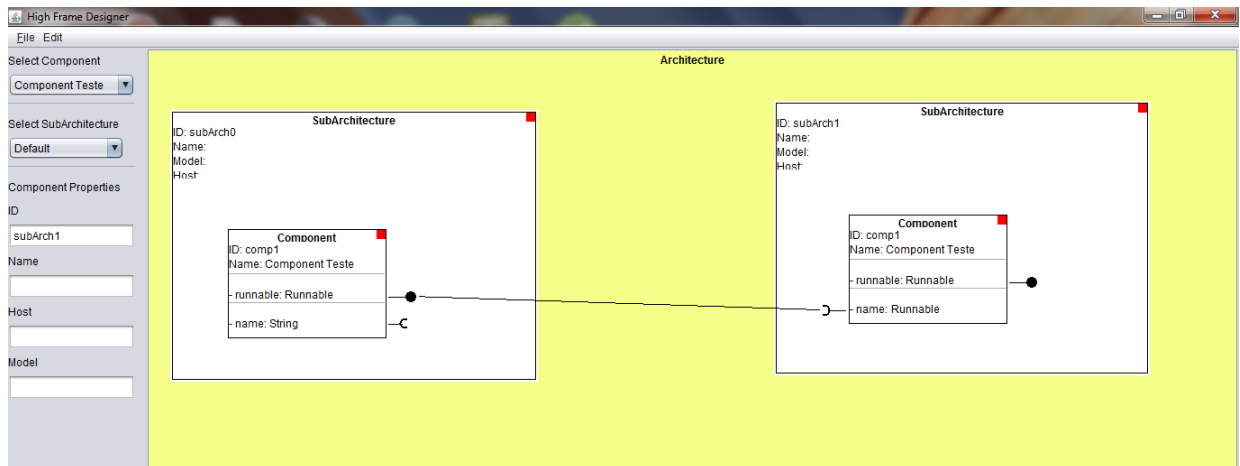


Figura 1.11: Bind entre subarquitecturas

1.5 Comanche WebServer

Apresentamos nesta seção um exemplo da composição de uma arquitetura distribuída com o HighFrame Designer. Este exemplo é baseado no *Comanche Web Server* e apresenta também componentes genéricos definidos com o HighFrame. O *Comanche Web Server* é um simples servidor web composto pelos seguintes componentes:

- *Receiver* - Componente responsável pelo recebimento das requisições HTTP de entrada;
- *Shceduler* - Componente responsável pelo escalonamento das requisições HTTP para análise;
- *Analyzer* - Componente que efetua a análise da requisições HTTP;
- *Logger* - Componente que registra as requisições de entrada;
- *Dispatcher* - Componente responsável pela interpretação das requisições HTTP de entrada;
- *FileHanler* - Componente que realiza a manipulação da solicitação de um arquivo;
- *ErrorHandler* - Componente responsável pelo tratamento de erros.

O uso de anotações para definição de componentes genéricos que serão transformados em componentes específicos é realizado através do modelo de programação Fraclet. Um exemplo do uso de anotações referente as interfaces fornecidas em um nível independente de modelo de componentes é apresentada no código fonte 1.2.

```

1 @Component(provides = @Interface(name = "r", signature = Runnable.class)
  )
2 public class RequestReceiver implements Runnable {
3
4     private final Logger log = getLogger("comanche");
5
6     private int port;
7
8     @Requires
9     private Scheduler s;
10
11    @Requires
12    private RequestHandler rh;
13 ...

```

Código Fonte 1.2: *Definição de interfaces fornecidas*

Neste código fonte o componente *RequestReceiver* foi definido através da anotação *@Component*. Este componente tem uma interface provida *Runnable*, demarcada através da anotação *@Interface*. Assim como, define duas interfaces requeridas através da anotação *@Requires*.

Com o HighFrame Designer foi possível montar uma arquitetura distribuído do Comanche WebServer conforme apresentado na figura.

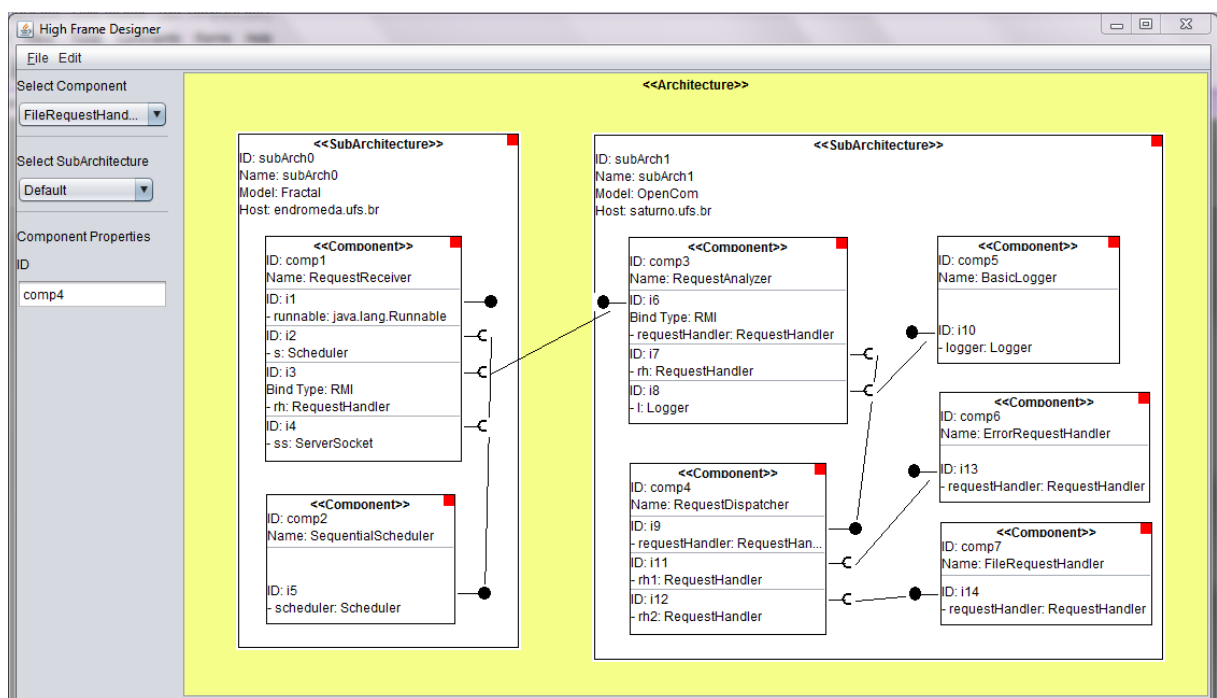


Figura 1.12: *Arquitetura distribuída do Comanche WebServer montada no HighFrame Designer*

O usuário monta a arquitetura do sistema distribuídos baseado em componentes em alto nível, não preocupando-se com particularidades de comunicação entre objetos distribuídos, *deployment* distribuídos e interoperabilidade entre componentes heterogêneos. Para o modelo apresentado podemos ter transformações de componentes para o modelo OpenCom em um nó e para o modelo Fractal em outro nó.