

Exercício 02 - Interface Gráfica para Cliente TCP

Autor: Felipe Fazio da Costa

RA: 23.00055-4

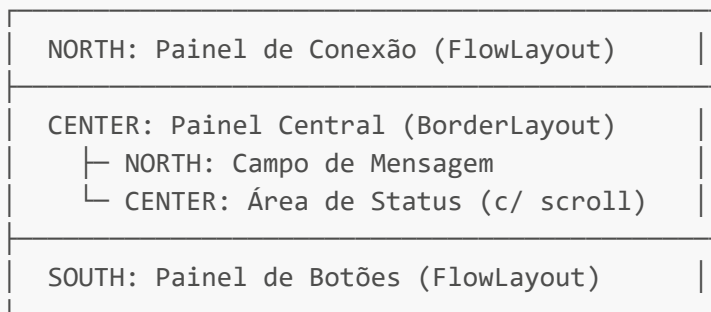
Descrição

Interface gráfica completa em **Java Swing** para o cliente TCP, substituindo totalmente a comunicação via console por uma interface visual intuitiva e funcional.

Componentes da Interface

Layout Principal: BorderLayout

A interface foi organizada usando **BorderLayout** para melhor distribuição dos componentes:



Componentes Específicos

1. Painel de Conexão (BorderLayout.NORTH)

- **Layout:** FlowLayout
- **Componentes:**
 - **TextField** `txtServidor` - Campo para endereço do servidor (default: "localhost")
 - **TextField** `txtPorta` - Campo para porta (default: "6789")
 - **Button** `btnConectar` - Botão para conectar/desconectar
 - Verde quando desconectado ("Conectar")
 - Vermelho quando conectado ("Desconectar")

2. Painel de Mensagem (BorderLayout.CENTER → NORTH)

- **Componente:** **TextField** `txtMensagem`
- **Funcionalidade:**
 - Campo para digitar mensagens a serem enviadas

- Desabilitado até estabelecer conexão
- Suporta tecla Enter para envio rápido
- Auto-limpeza após envio

3. Área de Status (BorderLayout.CENTER → CENTER)

- **Componente:** `JTextArea txtStatus` (somente leitura)
- **Funcionalidade:**
 - Exibe todo histórico da comunicação
 - Mostra mensagens do sistema
 - Mostra mensagens enviadas e recebidas
 - Auto-scroll para última mensagem
 - Scroll vertical sempre visível
 - Quebra automática de linha

4. Pannel de Botões (BorderLayout.SOUTH)

- **Layout:** `FlowLayout`
- **Componentes:**
 - `btnEnviar` (Azul) - Envia mensagem ao servidor
 - `btnLimpar` (Laranja) - Limpa histórico de mensagens
 - `btnSair` (Vermelho) - Encerra aplicação com confirmação

Funcionalidades Implementadas

**** Botão "Conectar/Desconectar" ****

Quando Desconectado:

- Valida campos de servidor e porta
- Estabelece conexão TCP com o servidor
- Cria streams de comunicação (InputStream/OutputStream)
- Atualiza `interface` para modo conectado
- `Habilita` campo de mensagem e botão enviar
- `Desabilita` campos de servidor e porta
- Muda texto para "Desconectar" e cor para vermelho

Quando Conectado:

- Envia comando `"-1"` ao servidor
- Aguarda confirmação `do` servidor
- Fecha streams e socket
- Atualiza `interface` para modo desconectado
- `Desabilita` campo de mensagem e botão enviar
- `Habilita` campos de servidor e porta
- Muda texto para "Conectar" e cor para verde

**** Botão "Enviar" ****

- Valida se está conectado
- Valida se há mensagem digitada
- Envia mensagem ao `servidor` (com quebra de linha)
- Aguarda resposta `do` servidor
- Exibe mensagem enviada e resposta no status
- Limpa campo de mensagem
- Retorna foco ao campo de mensagem
- Trata desconexão inesperada `do` servidor

Atalho: Pressionar `Enter` no campo de mensagem executa a mesma função.

**** Botão "Limpar" ****

- Solicita confirmação ao usuário
- Limpa toda área de status
- Limpa campo de mensagem
- Adiciona mensagem de confirmação da limpeza
- Mantém conexão `ativa` (se houver)

**** Botão "Sair" ******Se conectado:**

- Solicita confirmação com aviso de conexão ativa
- Desconecta graciosamente `do` servidor
- Fecha todos os recursos
- Encerra aplicação (`System.exit(0)`)

Se desconectado:

- Solicita confirmação simples
- Encerra aplicação

**** Fechar Janela (X) ****

- Captura evento `windowClosing`
- Executa mesma lógica do botão "Sair"

Tipos de Mensagens no Status

Mensagens do Sistema (Sistema:)

```
Sistema: Aplicação iniciada. Aguardando conexão...
Sistema: Conectando ao servidor localhost:6789...
Sistema: ✓ Conectado com sucesso!
Sistema: Desconectado do servidor.
Sistema: Histórico limpo.
Sistema: Encerrando aplicação...
```

Mensagens de Comunicação

```
Cliente → Servidor: Olá servidor
Servidor → Cliente: OLÁ SERVIDOR
```

Mensagens de Erro

```
Erro: Servidor não encontrado!
Erro: Não foi possível conectar ao servidor!
Erro: Servidor desconectou!
Erro ao enviar mensagem: Connection reset
```

Estados da Interface

Estado 1: Desconectado (Inicial)

Componente	Estado
txtServidor	✓ Habilitado
txtPorta	✓ Habilitado
btnConectar	✓ "Conectar" (Verde)
txtMensagem	✗ Desabilitado
btnEnviar	✗ Desabilitado
btnLimpar	✓ Habilitado
btnSair	✓ Habilitado

Estado 2: Conectado

Componente	Estado
txtServidor	✗ Desabilitado

Componente	Estado
txtPorta	✗ Desabilitado
btnConectar	✓ "Desconectar" (Vermelho)
txtMensagem	✓ Habilitado (Foco)
btnEnviar	✓ Habilitado
btnLimpar	✓ Habilitado
btnSair	✓ Habilitado

Validações Implementadas

Validação de Conexão:

- ✓ Servidor não pode estar vazio
- ✓ Porta não pode estar vazia
- ✓ Porta deve ser um número válido
- ✓ Trata servidor `inexistente` (`UnknownHostException`)
- ✓ Trata porta ocupada/inválida (`IOException`)

Validação de Envio:

- ✓ Verifica se está conectado
- ✓ Mensagem não pode estar vazia
- ✓ Trata desconexão durante envio
- ✓ Verifica se servidor `respondeu` (`null check`)

Validação de Saída:

- ✓ Confirma saída se conectado
- ✓ Avisa sobre conexão ativa
- ✓ Desconecta graciosamente antes de sair