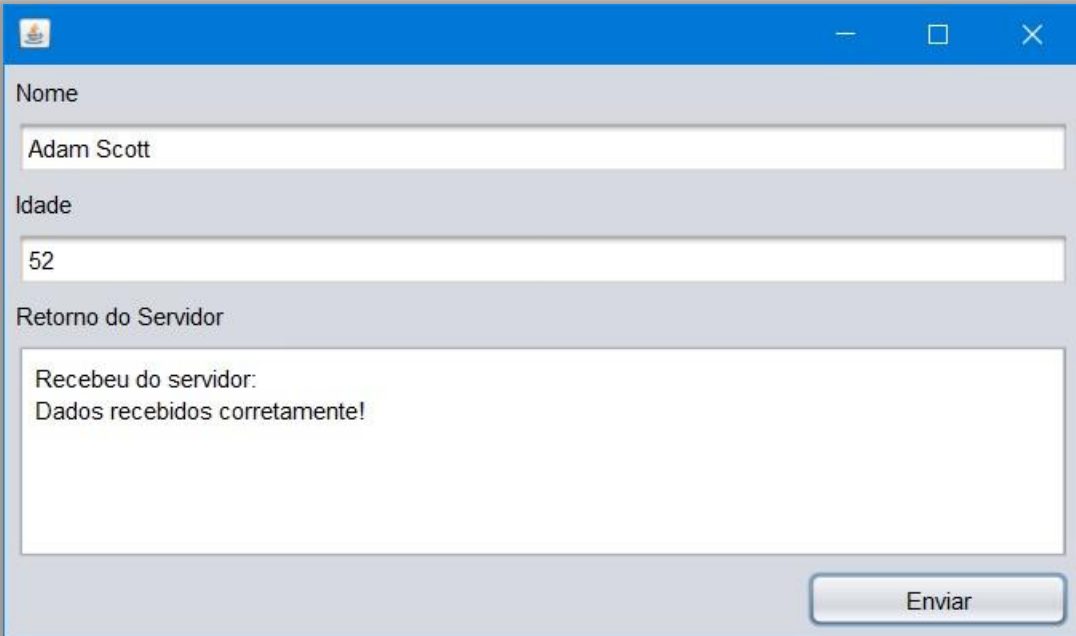


Enunciado da atividade

Desenvolva uma aplicação em que um cliente, utilizando componentes gráficos da biblioteca **Swing**, transmita o objeto **Pessoa** (atributos: *nome* e *idade*) para o servidor. O servidor deve exibir em seu console de execução, os dados recebidos do cliente. O cliente deve receber um aviso do servidor, informando que os dados foram transmitidos corretamente, mostrando isso em uma **TextArea**.

Utilize a figura abaixo como modelo para o cliente:

**Regras para construir o programa**

Utilize conexão do *localhost*, ou seja, o IP **127.0.0.1**, para a conexão cliente servidor. A porta deverá ser uma porta efêmera, **50000**. Este programa deverá ter a execução do servidor em **modo console**, e a execução do cliente em **modo gráfico**.

Não implementar nada mais do que o solicitado, tal como botões “*clique aqui para iniciar o servidor*” ou outras funcionalidades não solicitadas.

Lembre-se que o cliente não deve nunca iniciar o servidor. O servidor roda um serviço que é independente do cliente, **portanto o servidor deverá ser iniciado primeiro.**

- O seu projeto deverá ter **seu nome como autor**, em um comentário. A classe que não tiver seu nome como autor, renderá desconto global no exercício.

Exemplo:

```
// @author Nome e Sobrenome
```

- O seu projeto deverá ser entregue com a package **local.redes**

```
package local.redes;
```

O aplicativo que for entregue com *package* diferente, receberá desconto.

- O servidor, no **fim de tudo, deverá fechar a conexão.**
- **Não implementar Threads**, baseado em videoaula de apoio. **Threads** ficarão para a próxima atividade.

Dicas importantes para concluir esta atividade

Dica 01: Este exercício pode ser feito com apenas 3 arquivos. **Pessoa.java** (o objeto), **Servidor.java** (o servidor), uma classe **FormCliente.java** (trata toda a entrada do cliente e recebimento). Pode-se ainda dispor de um arquivo *.form* que é gerado pela **IDE**, para a reconstrução da interface. Dependendo da implementação o aluno pode ter que se amparar no uso do *.form* ou pode implementar sem o *.form*, ou leiaute gráfico.

Dica 02: Você usará *getters* e *setters* na sua classe **Pessoa.java**.

Dica 03: Lembre-se da criação de formulários na disciplina de **Linguagem de Programação Java I**. A IDE facilita muito a criação desses formulários, e depois, você incorpora as ações/eventos nesses campos/botões. Lembre-se dos métodos **actionPerformed** e do **AWT**.

Regras da Entrega

Anexe na atividade, os arquivos fonte **.java** e eventuais arquivos **.form**.

*Novamente: há alunos que conseguem desenvolver esse exercício sem a necessidade do arquivo **.form** gerado pela IDE, porém se não for o seu caso, não se esqueça desse arquivo.*

- **Servidor.java** (código fonte)
- **Pessoa.java** (código fonte).
- **FormCliente.java** (código fonte)
- Arquivo(s) adicional(is), se for o caso.

É permitido criar mais classes, porém, como já mencionado, **03 classes bastam** para construir esse pequeno programa.

Não compactar nenhum arquivo. O envio compactado, como é exigido nas outras disciplinas, **mas não nesta**, acarretará desconto.

Submeta **os arquivos** no ambiente de ensino.

Não deixar a tarefa em Modo Rascunho.

Clique no botão **Enviar Tarefa por Definitivo**.

Atenção para as penalidades no exercício

Quando o exercício recair em uma dessas situações e acumular erros, **os descontos abaixo serão aplicados**. A critério do avaliador, alguns descontos poderão ser isentados, dependendo do julgamento do prejuízo acumulado de nota.

Situação de descontos cumulativos	Desconto
Classes (ou uma delas) não possuem um comentário indicando sua autoria.	10,00
Entregou o exercício com <i>package</i> diferente.	15,00
Entregou o exercício em formato ZIP ou RAR ou outro, que é pedido nas outras disciplinas, mas não nesta .	15,00
Foi utilizada porta baixa ou porta reservada – o programa só funciona com a troca da porta.	20,00
Foi utilizado um IP de LAN – e não um endereço do escopo 127/8 – o programa só funciona com a correção para o <i>localhost</i> .	20,00
O programa não fecha o servidor após o final , bloqueando a porta do exercício, na correção do próximo aluno.	20,00
O aluno usou threads no programa, o que é proibido pelo enunciado, nesta atividade – independente se o uso é exemplificado em videoaula.	30,00
Servidor não retorna mensagem para cliente, mas o cliente envia dados para o servidor, com sucesso.	60,00
Somente servidor funciona e entra no estado bloqueante, cliente não compila ou nem funciona.	60,00

Existe a conexão entre o servidor e o cliente, mas o objeto não é passado para o servidor, logo, o servidor não responde.	60,00
O código não compila. O código é incompreensível.	100,00
Cópia de outro aluno, não importando se tudo está correto.	100,00