

Java Advanced

Prof. Dr. Marcel Stefan Wagner

Aula 16 – Criptografia de Chave Assimétrica e *Sockets*

FIAP

Criptografia de chave Simétrica e Assimétrica com Java

Aula 15

Prof. Dr. Marcel Stefan Wagner

Curso **Análise e Desenvolvimento de Sistemas**

FIAP – Faculdade de Informática e Administração Paulista

Campus – Lins / Paulista

Curso de **graduação** – 2025/2

Turma – **TDS**

Exercício de Criptografia de Chave Assimétrica via *Sockets*

Exercício 2

Construa uma aplicação em linguagem Java que funcione apenas no formato *Client-Server*, ou seja, com uma conexão via *sockets*, para realizar a criptografia (cifragem) de uma mensagem que deve ser inserida via terminal/console no cliente e enviada via *socket* para o servidor. O servidor, por sua vez, deve ser capaz de receber a informação criptografada e decifrar a mensagem.

O seu App deve ser capaz de receber no terminal/console do cliente, a informação da mensagem em forma de texto, realizar a criptografia de chave assimétrica, mostrando a informação criptografada no terminal/console, para depois enviar a mensagem criptografada para o servidor.

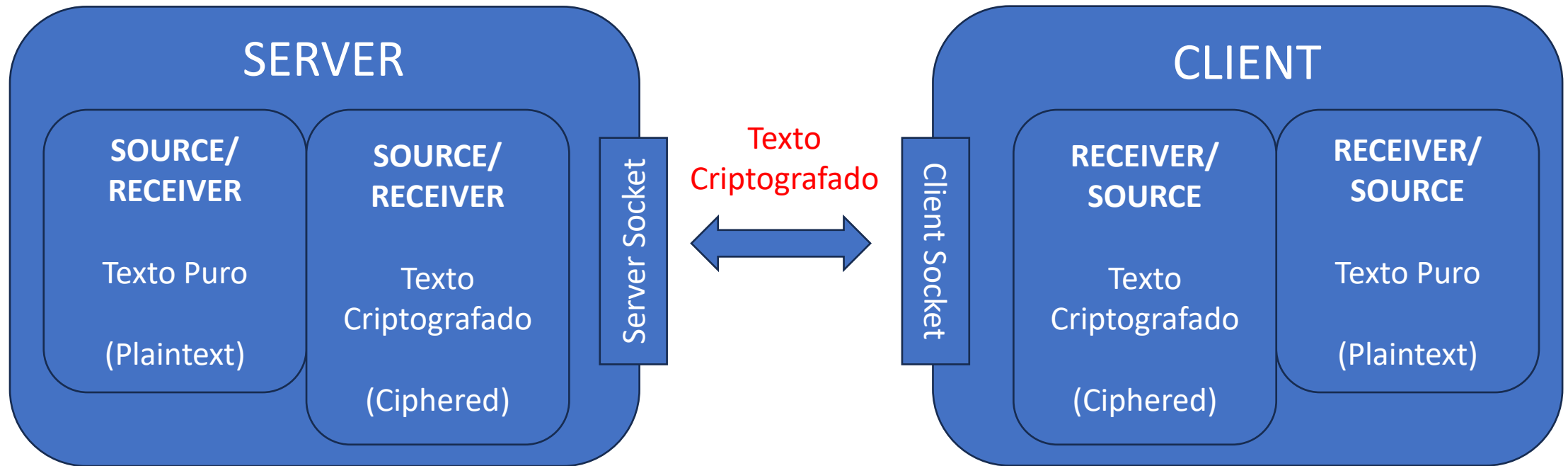
Exercício 2

O servidor, por sua vez, deve ser capaz de mostrar no seu terminal/console a mensagem criptografada que foi recebida, descriptografando a mensagem e mostrando a informação em *Plaintext* no terminal/console.

Na sequência o seu App deve ser capaz de realizar a comunicação contrária, ou seja, do servidor para o cliente, mantendo o mesmo padrão e tendo como característica a desconexão pelo lado do servidor.

Utilize os conceitos de *sockets* e de criptografia de chave assimétrica para desenvolver o seu aplicativo.

Exercício 2



Referências

George Coulouris, Jean Dollimore, Tim Kindberg, and Gordon Blair. **Sistemas Distribuídos: Conceitos e Projeto**. Bookman Editora, 5 edition, 2013.

Harvey M Deitel, Paul J Deitel, David R Choffnes, et al. **Sistemas Operacionais**. Pearson/Prentice Hall, 3 edition, 2005.

Maarten Van Steen and A Tanenbaum. **Sistemas Distribuídos: Princípios e Paradigmas**. Pearson/Prentice Hall, 2 edition, 2007.

Harvey M Deitel and Paul J Deitel. **Java, como programar**. Ed. Pearson/Prentice Hall, 8 edition, 2010.

StackOverflow. Disponível em: <<https://pt.stackoverflow.com/>>. Acesso em: abril de 2023.

GAMMA, Erich et al. **Elements of Reusable Object-Oriented Software**. Design Patterns, 1995.

COOPER, James William. **Java design patterns: a tutorial**. 2000.

GUERRA, Eduardo. **Design Patterns com Java: Projeto orientado a objetos guiado por padrões**. Editora Casa do Código, 2014.

Referências

AWS Amazon – O que é RESTful. Disponível em: <[https://aws.amazon.com/pt/what-is/restful-api/#:~:text=Representational%20State%20Transfer%20\(REST\)%20is,complex%20network%20like%20the%20internet.](https://aws.amazon.com/pt/what-is/restful-api/#:~:text=Representational%20State%20Transfer%20(REST)%20is,complex%20network%20like%20the%20internet.)>. Acesso em: maio de 2023.

InfoQ. Disponível em: <<https://www.infoq.com/minibooks/emag-03-2010-rest>>. Acesso em: maio de 2024.

Michelli Brito. Disponível em: <<https://www.youtube.com/@MichelliBrito>>. Acesso em: abril de 2024.

Obeautifulcode. Disponível em: <<http://blog.obeautifulcode.com/API/Learn-REST-In-18-Slides/>>. Acesso em: junho de 2022.

Berkeley. Disponível em: <<http://courses.ischool.berkeley.edu/i290-rmm/s12/slides/Lecture3%20REST.pdf>>. Acesso em: maio de 2023.

Trybe. Disponível em: <<https://blog.betrybe.com/tecnologia/deploy/>>. Acesso em: julho de 2024.

.....

Obrigado!

Agradecimentos ao professor:
Prof. MSc. Gustavo Torres Custódio

.....

Contato: profmarcel.wagner@fiap.com.br

Cursos:

Tecnologia em Análise e Desenvolvimento de Sistemas (TDS)

Tecnologia em Defesa Cibernética (TDC)

Engenharia de Software (ES)

