

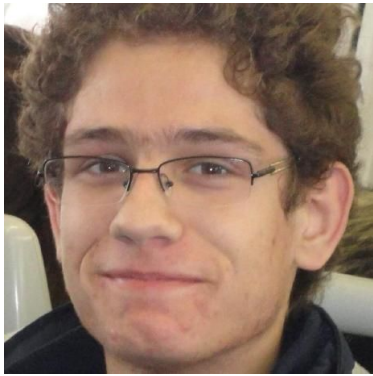


# Sistemas Distribuídos 15/16

---

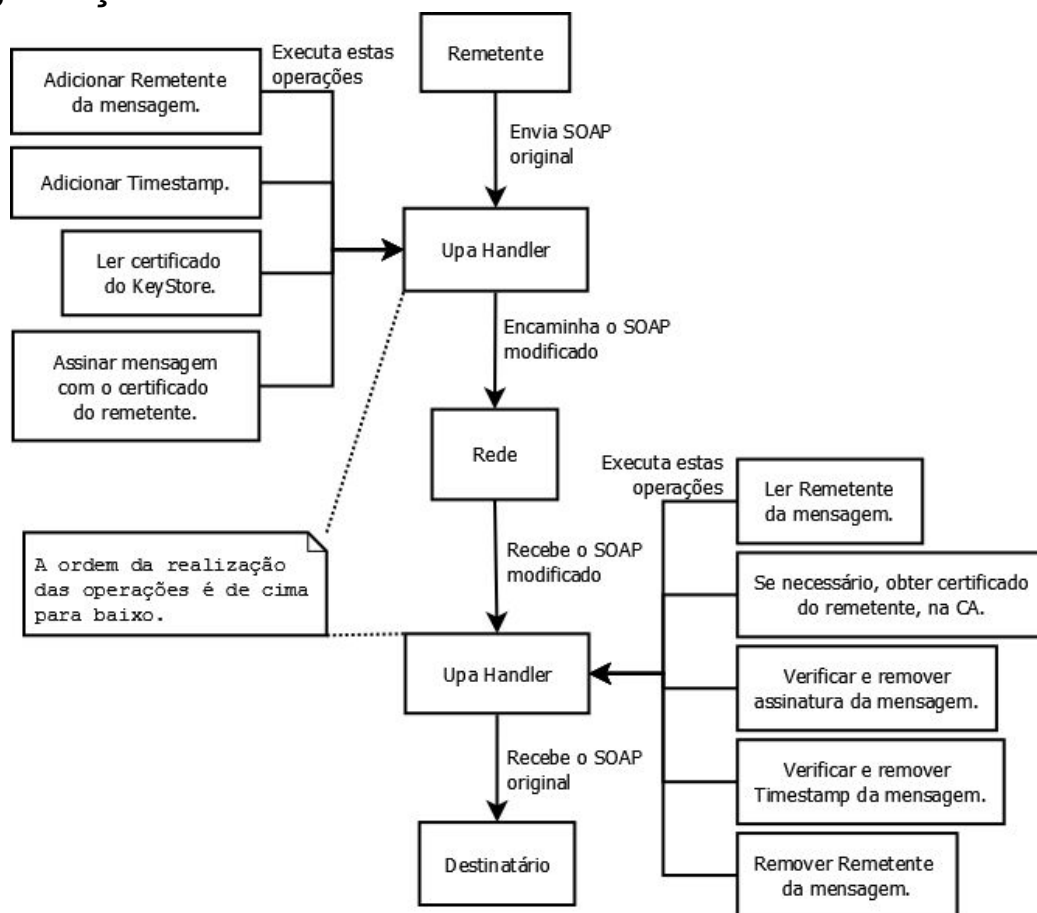
		
João Serras	Daniel Caramujo	Francisco Santos
79664	79714	79719

Grupo: A43

URL do Repositório:

[https://github.com/tecnico-distsys/A\\_43-project](https://github.com/tecnico-distsys/A_43-project)

## Segurança



O *UpaHandler* funciona em qualquer um dos sentidos da relação entre *Broker* e *Transporters*, por isso preferiu-se utilizar os nomes **Remetente** e **Destinatário**.

O funcionamento da solução é o seguinte:

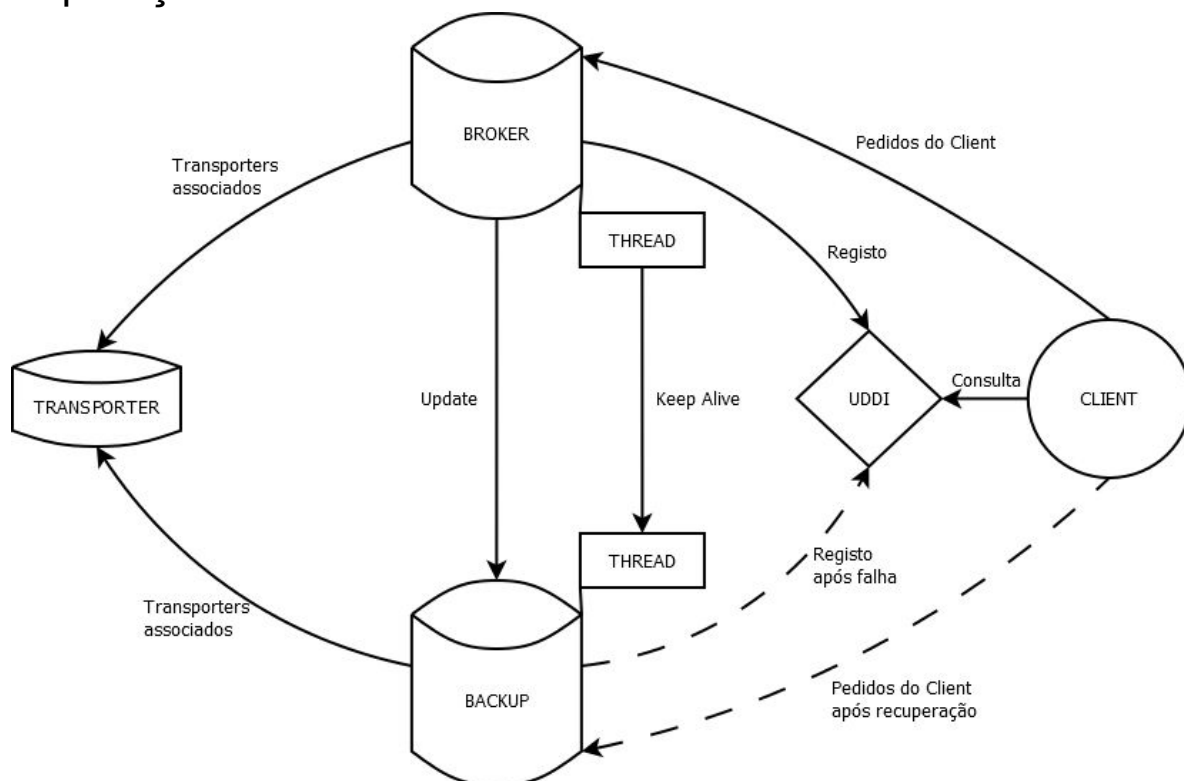
1. O **Remetente** ao ser iniciado inicializa uma variável, à qual o *handler* tem acesso, que define o seu nome.
2. A classe que trata da parte funcional do *Web Service* envia para o *handler* a mensagem SOAP como se fosse para enviar para o outro serviço.
3. O *handler* do **Remetente** recebe a mensagem:<sup>1</sup>
  - a. Adiciona o nome do remetente e o *Timestamp* ao *Header*;
  - b. Assina a mensagem usando a chave privada da sua *KeyStore*;
  - c. Verifica se a assinatura está correta e envia a nova mensagem SOAP modificada.
4. O *handler* do **Destinatário** recebe a mensagem:
  - a. Lê quem é o remetente e se necessário<sup>2</sup>, pede à CA o certificado deste;
  - b. Verifica se a assinatura bate certo, utilizando o certificado, e remove-a do *Header*;
  - c. Verifica se o *Timestamp* está dentro do tempo válido (1 minuto);
  - d. Retira o *Timestamp* e o remetente e envia a mensagem SOAP original ao **Destinatário**.

Em qualquer um dos passos de verificação, se houver alguma suspeita que a mensagem tenha sido proveniente de um ataque, é despoletada uma exceção.

<sup>1</sup> A descrição assegura a autenticidade, integridade, não repúdio e a frescura da mensagem.

<sup>2</sup> Certificado não presente, entrada não existente na tabela de contagens ou contagem de mensagens superiores ao definido.

## Replicação



Para permitir a tolerância a faltas e garantir um funcionamento contínuo e fiável do servidor *Broker* foi implementada uma solução que consiste na sua replicação. Um servidor *backup* recebe atualizações de toda a informação relevante modificada no servidor principal e confirmações de que este está operacional. A relação entre os dois servidores é a seguinte:

1. O *Broker backup* é iniciado, e fica à espera que exista um *Broker* que precise de um *backup*.
2. Iniciando de seguida o *Broker* principal, este estabelece uma ligação ao *backup*, que começa uma verificação do estado do servidor principal. A verificação consiste em mensagens enviadas constantemente pelo *Broker* principal (*Keep Alive*), e esperadas pelo *backup*. Caso essas mensagens não cheguem ao *backup*, este permite uma tolerância de três mensagens falhadas, e após esse limite o *backup* assume-se como *Broker* principal, mudando o seu nome no *UDDI*.
3. A cada operação feita no *Broker* principal, todos os dados relevantes são enviados para o *backup* associado (*Update*), de forma a garantir a consistência dos dados num caso de falha do servidor.
4. Caso haja falha do servidor principal o cliente tenta restabelecer a conexão ao servidor, um máximo de três vezes, com intervalos de alguns segundos entre cada tentativa, para dar tempo que o *backup* se assuma como principal ou que outros problemas de rede sejam resolvidos. Este não conhece a existência dos dois servidores, apenas procura o nome do servidor *Broker* no *UDDI*. Após ter sucesso continua a operação que estava a fazer, garantindo a continuidade do sistema.
5. Caso o *backup* do *Broker* falhe enquanto o *Broker* está a correr, as atualizações de informação param de ser enviados e é dada a informação que o *backup* deixou de existir. O *Broker* principal tem a capacidade de correr normalmente sem um *backup*.