

# Atividade de POS

Comunicação Assíncrona usando WebServices

## Controle de Versão

Versão	Data	Descrição	Responsável
1.0	29/03/2016	Criação do documento	Douglas

Para atender à problemática sugerida na atividade, foi desenvolvida uma aplicação, cuja arquitetura foi baseada no modelo desenvolvido pelo professor e que encontra-se representado na figura 1:

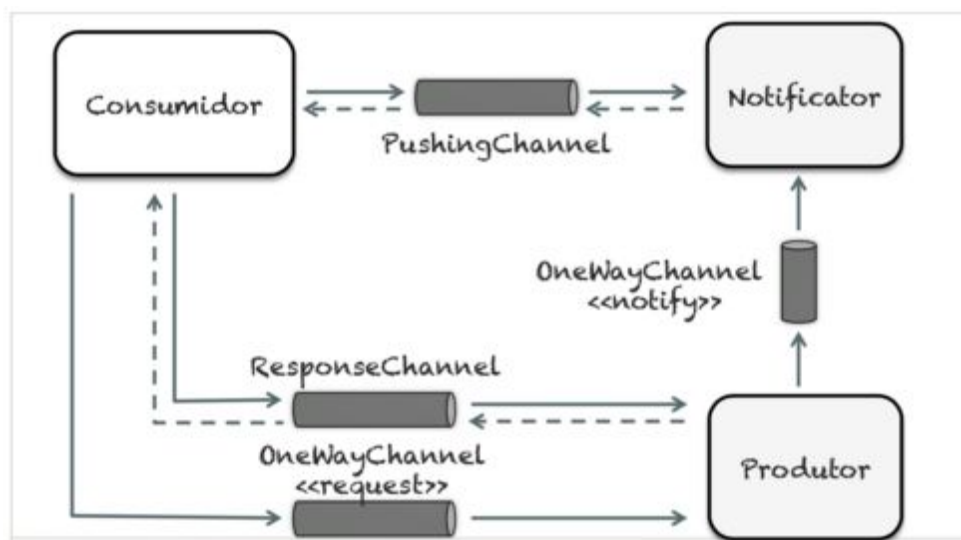


Figura 1: Representação arquitetural

Para maior detalhamento dos componentes desenvolvidos para atender as funcionalidades sugeridas, foi desenvolvido uma representação dos mesmos baseada no diagrama UML de componentes, conforme é observado na figura 2:

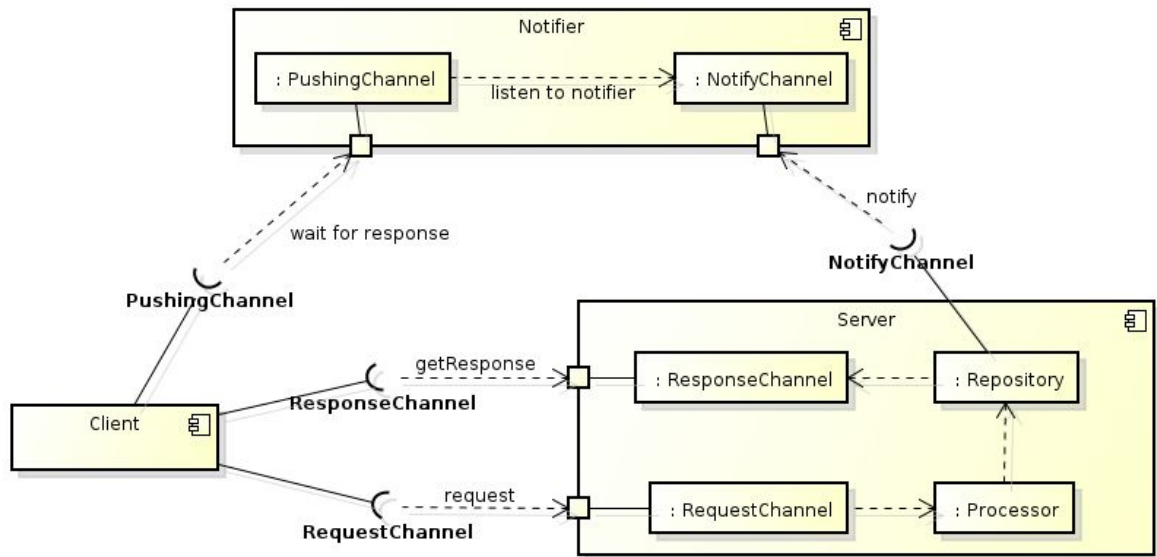


Figura 2: Representação dos componentes

É possível observar, portanto a divisão do componente **Server** em mais 4 partes, cada uma com suas devidas responsabilidades:

- **RequestChannel:** responsável por prover uma interface a qual o cliente pode solicitar o processamento de uma determinada mensagem. Para isto, é necessário que o mesmo identifique unicamente a mensagem enviada com uma senha (*id*), para que seja possível recuperá-la posteriormente;
- **ResponseChannel:** é o responsável por prover uma interface onde o cliente poderá solicitar a recuperação de sua mensagem processada, após um determinado período de tempo. Para este exemplo, o cliente apenas solicitará a recuperação da mensagem, quando for notificado;
- **Processor:** é o componente responsável por processar a mensagem informada pelo cliente. A título de exemplo, adiciona a mensagem "mensagem processada:" a frente da mensagem original, e para isso demora uma quantidade randômica de tempo;
- **Repository:** é o componente responsável por manipular as respostas dos clientes que solicitaram, além de utilizar-se do componente **Notifier** para notificar o cliente da conclusão do processamento;

Além do componente **Server**, o sistema ainda possui o componente **Notifier**, responsável por cuidar da notificação de clientes quando suas mensagens estiverem concluídas. Para isto, o mesmo é dividido em mais duas partes:

- **NotifyChannel:** é o componente responsável por prover uma interface onde o servidor poderá notificar seus clientes, apenas informando sua identificação;
- **PushingChannel:** é o componente responsável por estabelecer uma conexão persistente junto ao cliente, e informá-lo o momento em que sua mensagem está pronta. Para isto, o mesmo escuta as requisições de notificações no componente antes citado.

Por fim, há a presença do componente *Client*, responsável por consumir os serviços disponibilizados pelos demais. Para isto, o cliente deve gerar uma identificação única para a mensagem que deseja processar, e passá-lo junto com a mensagem. Após isto, o mesmo deve inscrever-se no componente *Notifier* para ser notificado quando sua mensagem for processada, para que, só assim, solicite novamente ao *Server* sua mensagem processada.

Para um melhor entendimento do fluxo de comunicação entre estes componentes, foi desenvolvido um diagrama de sequência que pode ser visualizado na figura 3:

