

# WebSphere MQ

*Bruno Miguel de Sousa Gonçalves*

## 1.Introdução ao WebSphere

Os produtos WebSphere providenciam comunicação entre programas através da interligação entre componentes heterogéneos, processadores, sistemas operativos, subsistemas e protocolos de comunicação. Trata-se de uma plataforma de integração e infraestrutura de aplicações da IBM.

Providencia funcionalidades avançadas através de diferentes módulos, por exemplo, WebSphere Application Server ou WebSphere Business Integration Server.

O Application Server trata-se de uma infraestrutura de aplicações e suporte a transacções. Desenvolvido em J2EE, uma API Java para desenvolvimento de aplicações empresariais, e serviços Web, oferece suporte fiável para aplicações Java e está disponível para um vasto número de plataformas. Está no cerne do desenvolvimento do WebSphere MQ.

O Business Integration Server providencia integração de processos e conectividade empresarial. Auxilia na criação de processos de negócio, sincroniza informação entre várias aplicações por várias plataformas e converte formatos de mensagens entre aplicações.

O WebSphere MQ permite ao utilizador trocar informação facilmente por diferentes plataformas, integrando aplicações novas com as já existentes.

Assegura a entrega fiável de mensagens, incluindo documentos XML e mensagens SOAP, interliga aplicações e serviços web, tem suporte para ambientes tais como J2EE, e reconhece um vasto número de plataformas (cisco, ibm...)

Utiliza SSL para comunicações seguras. As API's são suficientemente flexíveis para suportar a implementação e monitorização de standards locais.

Distribui dnâmicamente a carga pelos recursos disponíveis (gestão de carga de trabalho) e tem suporte para processamento transaccional.

Opcionalmente pode trocar mensagens em grupo usando filas transaccionais ou fazer segmentação de mensagens.

## 2.Arquitectura

O Websphere MQ permite que aplicações comuniquem através de mensagens e filas, geridas por gestores específicos (MQM's). Um gestor é responsável pela troca de mensagens entre interlocutores mas também executa algumas funções como triggering, onde aplicações são executadas de acordo com a chegada de mensagens específicas; coordenação de transacções; e conversão de tipos de dados.

As aplicações clientes ligam-se aos gestores através de um canal bidireccional fornecido pela API usada.

A troca de mensagens é feita começando por uma aplicação enviando uma mensagem para o canal, o gestor encaminha a mensagem para o remetente correcto, e por fim, o remetente lê a mensagem do seu canal.

Quando as aplicações são locais, as mensagens são entregues directamente à aplicação receptora, quando as aplicações estão ligadas a gestores diferentes, estes comunicam através de canais unidireccionais com informação acerca da plataforma que se encontra no extremo remoto e dos protocolos usados. Estes canais são usados em pares.

Para uma aplicação usar as funcionalidades do WebSphere, o programador poderá usar as API's fornecidas, mais concretamente, a AMI (Application Messaging Interface) que fornece um alto nível de abstracção; o JMS (Java Message Service), visto que o WebSphere é muito orientado ao uso de Java através do Application Server, e o MQI (Message Queue Interface) que é uma API de baixo nível, providencia chamadas nativas a funções de controlo das pilhas. As aplicações desenvolvidas nas três API's conseguem interoperar.

## 2.1.Application Messaging Interface

Para uma aplicação poder aceder às funcionalidades dos produtos WebSphere, o programador deve usar a Message Queue Interface na sua aplicação. A Application Messaging Interface providencia uma interface simples que os programadores podem usar sem necessitar de conhecer todas as funções disponíveis na MQI. As funcionalidades disponibilizadas para uma determinada aplicação, são definidas por um administrador usando serviços e políticas, descritas mais a frente.

Existem três conceitos importantes na AMI:

- .A *mensagem*, que define o que é enviado de um programa para outro,
- .O *serviço*, que define por onde ou para onde a mensagem é enviada,
- .A *política*, que define como a mensagem é enviada.

Para enviar uma mensagem, a aplicação deve definir o tipo de dados, o serviço e a política a usar. É possível usar serviços e políticas por omissão, ou criar outras, que opcionalmente podem ser guardadas num repositório.

Existem 4 formas de enviar uma mensagem:

- ."Sent and forget" – uso de datagramas, quando não é necessário reply.
- .Listas de distribuição – quando se quer enviar uma mensagem para vários destinatários, o administrador pode criar listas de distribuição
- .Pedido/resposta – quando se espera por uma resposta ao pedido.
- .Publish/Subscribe – quando existe um broker a fazer a distribuição de mensagens.

A disponibilidade de cada forma depende da configuração do sistema.

As mensagens consistem em atributos e dados, alguns dos atributos de mensagem são:

- .MessageID – identificador da mensagem
- .Format – estrutura da mensagem
- .Topic – indica o conteúdo da mensagem para aplicações Publish/Subscribe
- .CorrelID – identificador que serve para correlar mensagens de resposta a mensagens de pedido

Usando a MQI directamente para programar uma aplicação, estes atributos são definidos explicitamente pelo programador, na AMI são contidos no objecto da mensagem, ou definidos numa política.

Os serviços representam destinos para onde as aplicações enviam mensagens ou de onde recebem respostas. No WebSphere, estes destinos representam as filas de mensagem, que residem em gestores de filas (MQM's). Os programas podem usar as API's para colocar mensagens nas filas ou retirar mensagens de lá.

Existem diferentes tipos de serviço para especificar o mapeamento da AMI para recursos reais na rede:

- .Túneis de comunicação unidireccionais entre aplicações com início na fila de envio do emissor e terminação na fila de recepção do receptor.
- .Listas de distribuição, contendo a lista das filas das aplicações para onde as mensagens devem ser enviadas.
- .Aplicação fornecedora com um módulo emissor usado para publicar mensagens num broker publish/subscribe
- .Aplicação subscritora com um módulo emissor usado para subscrever um broker publish/subscribe, e um receptor para receber publicações do broker.

A AMI tem funções para abrir ou fechar serviços explicitamente, mas estes também podem ser abertos ou fechados implicitamente por outras funções. O uso de um serviço depende claro da configuração do sistema, por exemplo, túneis de comunicação apenas fazem sentido em configurações que usem "Send and forget" para enviar as mensagens, e módulos emissores e receptores em aplicações fornecedoras/consumidores fazem sentido quando existe um broker de mensagens

As políticas controlam o modo como as funções da AMI funcionam. Alguns dos itens controlados pelas políticas são:

- Atributos que constituem mensagens
- Número máximo de filas por gestor
- Número máximo de ligações a um gestor
- Permissões de acesso a objectos (filas e listas de distribuição)
- Definição de gestores por omissão

A AMI pode interoperar com outras API's, logo as aplicações desenvolvidas com ela podem trocar mensagens com:

- .Outras aplicações que usem AMI
- .Qualquer aplicação que use a MQI

.Um message broker, como por exemplo o WebSphere MQ Publish/Subscribe ou o WebSphere MQ Integrator, não discutidos nesta literatura.  
Está disponível nas linguagens de programação C, COBOL, C++ e Java.

Algumas das funções disponibilizadas pela AMI são:

MQCONN – liga uma aplicação a um MQM, devolve um handler para a ligação que é usado posteriormente para o envio e recepção de mensagens.

MQOPEN – estabelece acesso a um objecto. Os seguintes tipos de objectos são válidos:

- .Filas e listas de distribuição
- .Listas de nomes
- .Definição de processos
- .Gestores de filas

MQPUT – coloca uma mensagem numa fila ou numa lista de distribuição. Qualquer uma delas tem que estar previamente aberta pela chamada MQOPEN.

MQGET – lê uma mensagem de uma fila que foi previamente aberta pelo MQOPEN.

MQCLOSE – fecha uma ligação a um objecto aberto pelo MQOPEN.

MQDISC – fecha a ligação com o MQM aberta com o MQCONN

## **2.2.WebSphere Application Server**

Desenvolvido em J2EE, o Application Server funciona em conjunto com um servidor HTTP para tratar pedidos de conteúdo dinâmico por parte de aplicações Web.

O servidor HTTP e o Application server comunicam com o plug-in WebSphere HTTP para o servidor HTTP. O plug-in usa um ficheiro de configuração em XML para determinar se um pedido deve ser tratado pelo application server ou pelo servidor HTTP.

É possível definir vários servidores, cada um executando na sua própria JVM.

Existem vários containers que se podem acoplar ao application server. A função destes containers é providenciar suporte para tipos de aplicações específicos.

-EJB Container – Suporta aplicações EJB.

EJB trata-se do componente server-sided para a plataforma Java 2.

Processo servidor que trata pedidos para session ou entity beans. Em conjunto com o servidor, o EJB providencia o BRE (Bean Runtime Environment). Providencia serviços de baixo nível, como threading e suporte a transacções.

- Web Container – Suporta pedidos de clientes HTTP (Browsers). Providencia servlets, JSP (JavaServer Pages) e outros tipos de ficheiros server-sided. Também podem gerar dados formatados, por exemplo em XML, para serem usados por outros componentes.
- Application client Container – Aplicações cliente são programas Java que correm tipicamente num computador desktop com uma GUI. Este container trata pedidos destas aplicações para várias fontes, como por exemplo, JDBC (Java Database Connectivity) ou filas pertencentes ao JMS (Java Message Service).
- Applet Container – As applets são muitas vezes usadas em conjunto com HTML, podendo tratar algum trabalho de processamento do servidor no cliente. O Applet Container trata pedidos vindos de applets embutidas em documentos HTML.
- HTTP Server Embutido – O Application Server suporta ligações de clientes HTTP. O cliente liga-se a um servidor Web que, usando o plug-in HTTP pode reencaminhar os pedidos para o Application Server.
- Virtual Host – Trata-se de uma configuração que possibilita a um host único, emular vários hosts. Apesar de poderem estar na mesma máquina física, não é possível partilharem dados. Um administrador pode usar um virtual host para associar uma aplicação Web específica a um host particular configurado para tratar aquela aplicação.

### **2.3.WebSphere Business Integration Server**

O Business Integration Server é usado a um nível superior ao do Application Server. Oferece uma base para integração das aplicações empresariais, sincroniza informação entre vários pontos, e converte dados entre as várias aplicações.

Estas funcionalidades são suportadas por adaptadores. Alguns adaptadores disponíveis são:

- Application Adapters – Extraem dados e informação de transacções de várias aplicações proprietárias ou empresariais e ligam-nas ao um hub central onde a informação é trocada.
- Mainframe Adapters – Utiliza tecnologias para aceder a dados de aplicações a correr em sistemas OS/390 e OS/400 (sistemas operativos da IBM para mainframes).

-Technology Adapters – Providencia conectividade entre vários protocolos e tecnologias melhorando a infraestrutura de integração.

-Adapter Development Tools – Conjunto de ferramentas para desenvolvimento de Adaptadores.

### **3.Vantagens e desvantagens do MQSeries.**

O MQSeries tem reputação de ser robusto e fiável.

A sua API é simples e as pilhas de mensagens podem ser partilhadas.

A IBM tem alguns problemas a aceder a tecnologias da concorrência de forma a suportar ligações entre as suas próprias e as da concorrência.

Um administrador necessita de bastante formação, pois as funcionalidades variam entre plataformas.

O preço varia consideravelmente entre diferentes plataformas, logo é necessário ponderar o uso de MQSeries caso os componentes sejam demasiado caros para as plataformas disponíveis.