

# Microsoft BizTalk Server aos olhos dos programadores

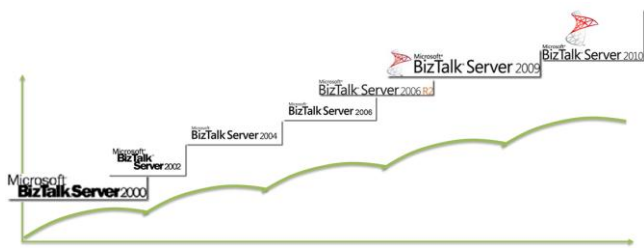
Já muito se tem falado sobre a plataforma BizTalk Server, o que é, e as vantagens que oferece as organizações. Para os mais distraídos, BizTalk Server é a plataforma de excelência da Microsoft para a integração de sistemas e processos empresariais.

Mas quais os benefícios que esta plataforma oferece aos programadores?

## Mercado de trabalho

Onde o produto é utilizado? Em que contextos? Quais as oportunidades de trabalho que oferece? Remunerações? Estas são algumas perguntas que um programador quer saber antes de se dedicar a uma tecnologia/produto.

**Microsoft BizTalk Server 2010** é a sétima versão do produto, tornando-o num dos produtos mais maduros e estáveis da Microsoft.

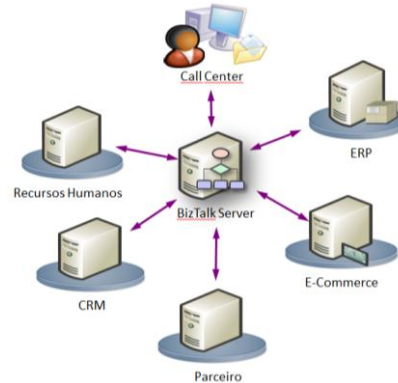


Ao longo dos anos o produto tornou-se uma referência de sucesso entre os sistemas de integração, sendo actualmente o produto mais utilizado no mundo na sua área, como asseguram os seguintes factos:

- Mais de 10 000 clientes espalhados por todo o mundo
- 81% do "Top 100" da revista Fortune
- 12 dos 15 maiores retalhistas mundiais utilizam BizTalk
- 5 das 10 maiores cadeias de hotéis mundiais utilizam BizTalk
- 9 das 10 maiores seguradoras mundiais utilizam BizTalk
- 23 dos 27 governos membros da UE utilizam BizTalk para prestação de serviços do governo
- 9 das 10 maiores companhias de telecomunicações norte americanas utilizam BizTalk
- 6 das 8 maiores companhias farmacêuticas norte americanas utilizam BizTalk
- 9 das 10 maiores companhias espaciais e de defesa nos EUA utilizam BizTalk

Com uma robusta infra-estrutura de mensagens, funcionalidade de resiliência (*Dehydration* e *Rehydration*), mais de 25 adaptadores multiplataforma, motor de regras, possibilidade de obter informações de desempenho sobre processos críticos de negócios, *debug*, persistência, tratamento e recuperação de erros, suporte a transacções.

Torna o BizTalk Server numa ferramenta e infra-estrutura única, ideal para ser usado principalmente para integração de aplicações corporativas (EAI), integração de sistemas entre parceiros de negócio (B2B) e para gestão de processos de negócio (BPM).



No que diz respeito às oportunidades de trabalho, podemos definir 3 perfis:

**BizTalk Architect:** conhecedor de todo o sistema de integração, *backup* e planos de recuperação, segurança, *logging*, o fluxo de mensagens, interface de comunicação. Utilizando os seus conhecimentos também como um programador, o arquitecto deverá conhecer as capacidades e limitações das ferramentas à sua disposição (BizTalk, BAM, BRE) e desenhar os projectos fazendo as melhores escolhas para cada situação.

**BizTalk Developer:** implementa e estende as funcionalidades base, tirando partido das diferentes ferramentas. Aqui existem muitas áreas completamente ortogonais e um programador poderá não dominar ao mesmo nível todas elas: *Orchestration*, *Adapters*, *Pipelines*, *Mappings*, *Functoids*, *Routing*, *Rules*, *Tracking*, *OLAP*, entre outras.

**BizTalk Administrator:** um administrador de sistemas terá outras preocupações, como sejam a saúde dos servidores e a sua actividade (HAT), desbloqueando mensagens e processos, garantindo o devido fluxo das mensagens, bem como o fluxo de dados de telemetria necessários ao bom diagnóstico dos processos de negócio.

Apesar do nosso mercado interno ser muito diferente do internacional aqui estão alguns exemplos reais de oportunidades de trabalho (fonte *LinkedIn*):

- "BizTalk Developers needed in": Miami, NYC, Jacksonville, Austin, Sacramento, Louisville, Hawaii, Panama City, Montevideo, London, Toronto, Preston, Porto.

# Microsoft BizTalk Server aos olhos dos programadores

- “BizTalk Administrator needed in Ft. Lauderdale FL 75K - 100K depending on experience.”
- “BizTalk Developer/Architect needed for International Law Firm in NY. 110k-130k.”
- “6 Month BizTalk 2009 Consultant required for Public Sector project in London. Rates negotiable but c£500-£550 per day.”
- “2 BizTalk Developer Needed in NYC - Healthcare, mid-level \$95-110K and senior level \$105-120K.”

## Funcionalidades para os programadores

Não há nada que o BizTalk faça que com código escrito de raiz não se consiga fazer, a questão é quanto tempo demora a implementar uma solução sem um uso de *middleware* (sistema de integração). Facilmente se conseguirá efectuar uma integração entre dois sistemas com o uso de tecnologias de comunicação e transporte de dados tais como o Microsoft WCF, mas quando se analisa com mais atenção alguns requisitos, é fácil identificar certos desafios que se podem tornar bastante complexos: implementação de persistência, correlação de mensagens assíncronas, recuperação em caso de falhas nas comunicações, mapeamentos complexos, atomicidade, transacções de longa duração, monitorização e visibilidade dos processos, e tudo isto se reflecte, para um programador, em custos enormes na implementação.

É aqui que o BizTalk entra e o que ele faz de melhor, porque todas estas funcionalidades estão ao dispor dos programadores “out of the box” com o produto, libertando assim, os mesmos, da necessidade de terem de reimplementar vezes sem conta estas funcionalidades e permitindo assim um foco no aspecto mais crítico que é a implementação da lógica de negócio associado aos fluxos de integração.

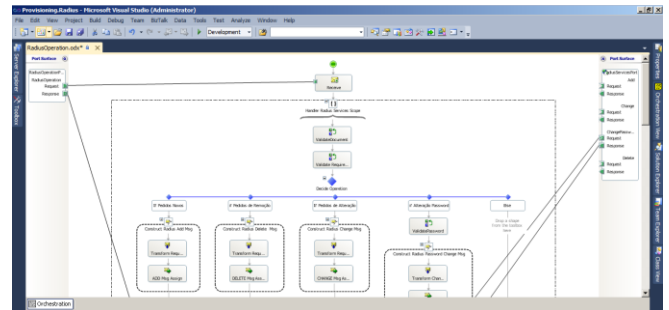
São diversas as funcionalidades ou módulos que poderíamos enumerar. Umas ajudam a simplificar a interoperabilidade, outras a reduzir custos na implementação:

### BizTalk Orchestration Designer

Antes de aprender a programar é ensinado a todos os programadores como representar todos os passos necessários para a execução de um processo na forma de um fluxograma, isto porque é mais legível e ilustra de forma simplificada o progresso da execução.

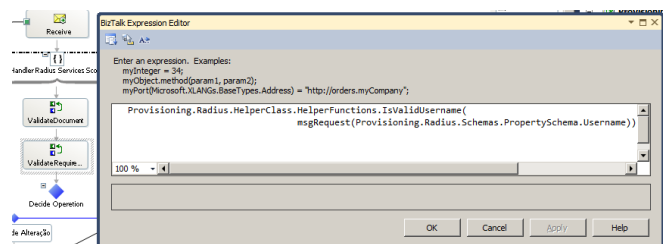
O BizTalk inclui um *Orchestration Designer*, integrado no Visual Studio, que possibilita aos programadores

representarem os processo de negócio de forma visual, assim como uma representação das portas, configurações e das ligações entre as diversas actividades (*shapes*), tornando-se assim mais fácil de gerir e ler do que numa linguagem textual generalista (exemplo C#).



### BizTalk Expression Editor

O *BizTalk Expression Editor* permite aos programadores introduzir código .NET, com suporte a *intellisense*, directamente nas orquestrações ou invocar bibliotecas externas, o que em certos cenários se torna bastante útil, como por exemplo: manipular valores das mensagens a partir da **Message Assignment** shape, manipular variáveis na **Expression** shape, construção de expressões booleanas dentro de **Loop** e **Decide** shapes, definir tempo de pausa na **Delay** shape, ou até mesmo configurar portas dinâmicas de saída.



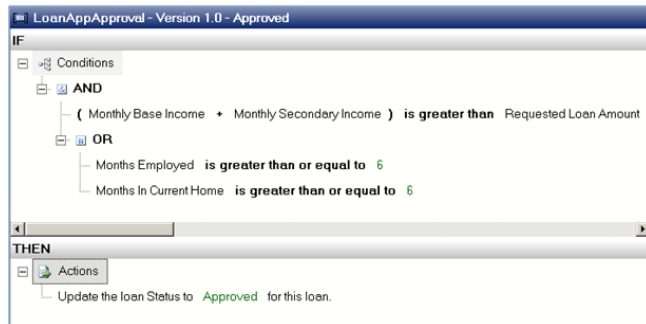
### Motor de regras (BRE)

O BizTalk Server inclui o *Business Rules Framework* onde podem ser isolados critérios de decisão de negócio. Sendo que os módulos principais incluem o **Business Rule Composer** para construção de políticas (regras), o **Rule Engine Deployment Wizard** para as instalar no *Run-Time Business Rule Engine*.

A tarefa da criação e alteração das regras poderá não ser uma tarefa do programador. Os analistas, consultores ou responsáveis de negócio poderão criá-las e a qualquer momento actualizar as mesmas. O programador poderá então reutilizar as regras de negócio nas suas orquestrações para suportar uma variedade de cenários

# Microsoft BizTalk Server aos olhos dos programadores

como por exemplo para determinar o caminho de execução de um processo de negócio, ou um valor a aplicar numa transacção. Mais que um parâmetro aplicacional, este motor permite que o valor ou condição seja completamente dinâmico e apenas determinado em contexto de *runtime*.

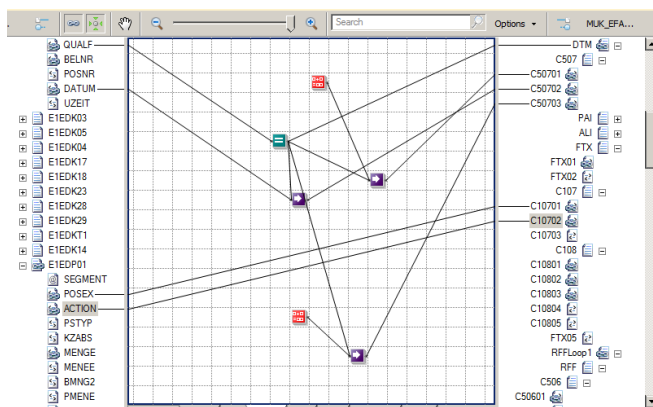


Usar regras, que mudam constantemente, em vez código poderá permite aos programadores não terem de refazer sistematicamente as suas aplicações.

Para mais informação: [The Business Rules Framework](#)

## BizTalk Mapper Designer

Inclui o *BizTalk Mapper Designer*, integrado no Visual Studio, que possibilita efectuar transformações de mensagens complexas de forma visual e extremamente simples.



Na realidade, este editor está a gerar um mapa XSLT, e em determinadas situações poderá ser mais adequado usar *snippets* XSLT directamente no mapa. Estas opções dependem muito da experiência do programador. Em termos de estruturação/modularidade, estes mapas podem ainda ser enriquecidos com funções (*Functoids*), chamadas externas (SQL *lookups*, outros mapas) ou código XSLT, .NET/C#, COM, VBscript.

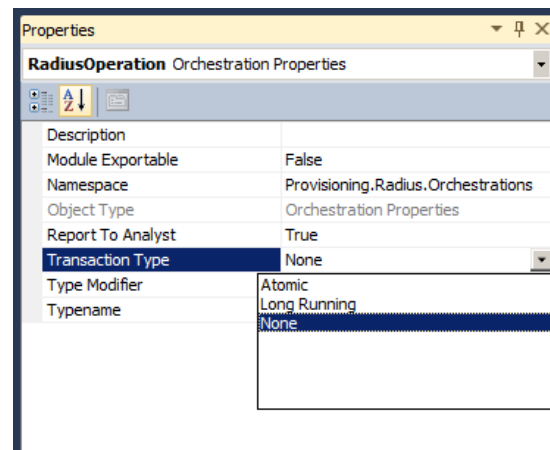
## Transacções, Excepções e Persistência de dados

É comum nas aplicações *Service-Oriented Architecture* (SOA) ou *Business Process*, onde os processos podem abranger vários *endpoints* (sistemas), as operações não poderem ser realizados num curto período de tempo, o que origina a que as transacções ACID não sejam as mais adequadas para este tipo de cenários.

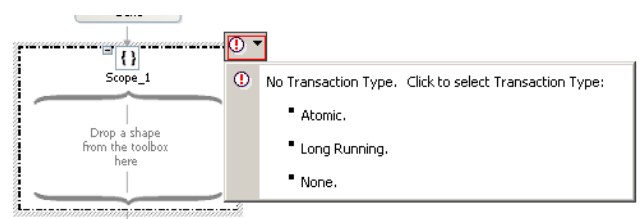
Ao contrário da programação tradicional, o BizTalk Server suporta dois tipos distintos de transacções:

- **Atómicas:** permite que uma transacção volte automaticamente para o seu estado anterior em caso de a operação não seja concluída com sucesso. Podemos desenhar orquestrações com suporte ACID (*Atomic*, *Consistent*, *Isolated* e *Durable*) configurando a *Scope* (ou mesmo ao nível da orquestração) como atómica.
- **Longa duração:** Estes processos podem ficar activos por dias, semanas ou por períodos de tempo mais longos, podem conter várias transacções (*nested transactions*), e possibilita tratar excepções para recuperação de falha. Suporta consistência e durabilidade.

Entre várias opções, pode-se configurar o tipo de transacção ao nível da orquestração:

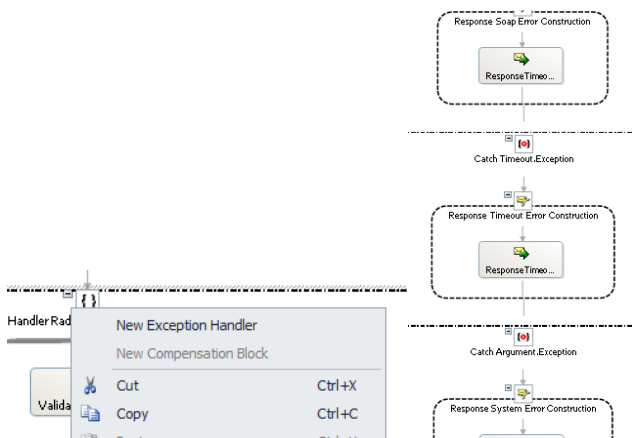


Assim como definir a transacção ao nível do “*scope*” por forma a encapsular uma unidade de trabalho dentro de um contexto de transacção:



# Microsoft BizTalk Server aos olhos dos programadores

O programador também tem a sua disposição a possibilidade de definir vários blocos (*Handlers*) para tratamento de excepções:



Quando lidamos com processos de negócio, nomeadamente os de longa duração, necessitamos sempre de falar de persistência dos dados por forma a salvaguardamos de falhas e podermos efectuar o reprocessamento a partir de uma determinado ponto do processo. A *Orquestration engine* persiste, automaticamente, o estado das instâncias em execução baseado no desenho das orquestrações efectuada pelo programador, existindo alguns eventos ou etapas em que é despoletado a operação de persistência, chamados pontos de persistência:

- **Ao nível do BizTalk Engine:** quando as instancias das orquestrações são suspensas, quando o sistema é desligado de forma controlada, quando a *engine* determina hidratar/desidratar os processos ou quando uma instância da orquestração é concluída.
- **Ao nível da Orquestração:** quando o final de uma *scope* transaccional é alcançado, na execução de outras orquestrações através da *Start Orchestration shape*, na *Send Shape* ou em *breakpoints* de *debugging*.

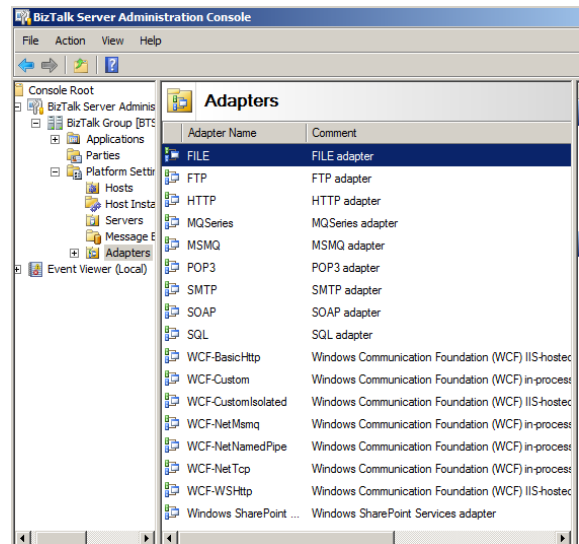
A persistência do estado das instâncias inclui: o progresso actual da instância, o estado de qualquer componente .NET que contém informação do estado e está a ser utilizado na orquestração assim como os valores das mensagens e variáveis.

Para mais informação:

[BizTalk Orchestration – Understanding Persistence points](#)

## Adaptadores

Inclui mais de [25 adaptadores](#) que simplificam a integração directa para as *top “Line of Business (LOB) Applications”* (tais como o Siebel, SAP, JD Edwards, Oracle ou Dynamics CRM), Base de dados (Microsoft SQL Server, Oracle, DB2) e outras tecnologias (Tibco, Java EE).



## WCF LOB Adapter SDK

Fornece um modelo de programação enriquecido para desenvolvimento de adaptadores baseados em *Windows Communication Foundation*. ([WCF LOB Adapter SDK](#))

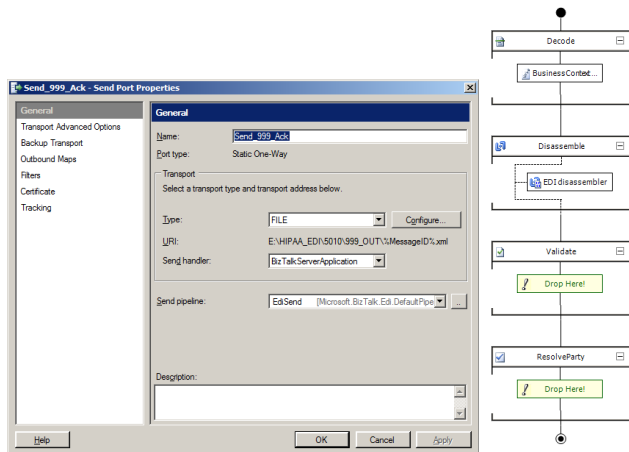
## Suporte poderoso e completo para EDI/AS2

*Electronic Data Interchange (EDI)* é uma das formas mais comuns de comunicação electrónica entre organizações (Facturas, encomendas, notas de débito). Estes padrões (EDIFACT, ASC X12) são projectados para leitura electrónica, e portanto inadequados para leitura humana, reflectindo à troca estruturada de dados de negócios entre sistemas utilizando um formato de dados padronizado que garante a fiabilidade dos dados através de diferentes *checksums*.

- As aplicação BizTalk EDI contém *pipelines*, orquestrações e *schemas* que são úteis e indispensáveis para a capacidade de processar transacções de EDI.
- A *pipeline* de entrada EDI é capaz de dividir os lotes EDI (*EDI Batch*), analisar os documentos, converter os arquivos EDI em arquivos XML e efectuar validações EDI e/ou XSD.
- A *pipeline* de envio EDI permite converter arquivos XML em documentos X12 ou EDIFACT, codificar as mensagens e efectuar validações XSD e/ou EDI.



# Microsoft BizTalk Server aos olhos dos programadores



- A *Trading Partner Management Interface* permite que uma organização defina as propriedades de processamento para os parceiros comerciais fazendo intercâmbio EDI.

The screenshot shows the 'Agreement1 - Agreement Properties' dialog box with the 'Envelopes' tab selected. It displays a table of envelope settings for various transactions.

| Transaction Type *     | Version/Release * | Target namespace *      | GS1     | GS2       | GS3       | GS4        | GS5       | GS7    | GS8        |
|------------------------|-------------------|-------------------------|---------|-----------|-----------|------------|-----------|--------|------------|
| 277Health Care Info... | 00401             | http://schemas.micro... | HNH...  | BCG123... | APH123... | CCYYMDD... | HHMMSS... | XAc... | 00401      |
| 999Implementation...   | 00501             | http://schemas.micro... | FAFu... | BCG123... | APH123... | CCYYMDD... | HHMMSS... | XAc... | 005010291A |
| 837_PHealth Care C...  | 00401             | http://schemas.micro... | HC.H... | BCG123... | APH123... | CCYYMDD... | HHMMSS... | XAc... | 00401      |
| 837_IHealth Care C...  | 00401             | http://schemas.micro... | HC.H... | BCG123... | APH123... | CCYYMDD... | HHMMSS... | XAc... | 00401      |

Para mais informações:

[EDI Support in BizTalk Server](#)

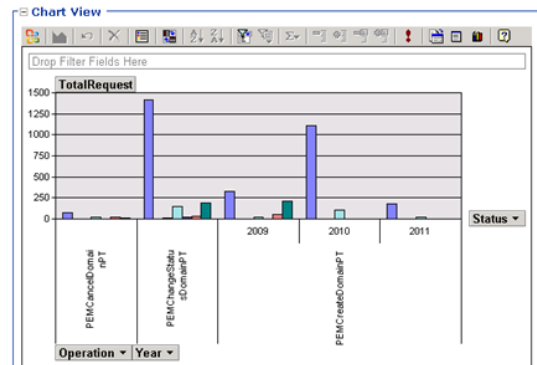
[EDI Support in BizTalk Server 2000, ..., 2006, 2009, and 2010](#)

## Monitorização de processos de negócio (BAM)

O *Business Activity Monitoring* ou BAM é um módulo que captura dados de negócio e *milestones* do processo permitindo que os analistas de negócios monitorizem e analisem os dados em tempo real.

The screenshot shows the 'Pivot Table View' with a dynamic table titled 'Tabela dinâmica2'. The table displays request data categorized by operation and year.

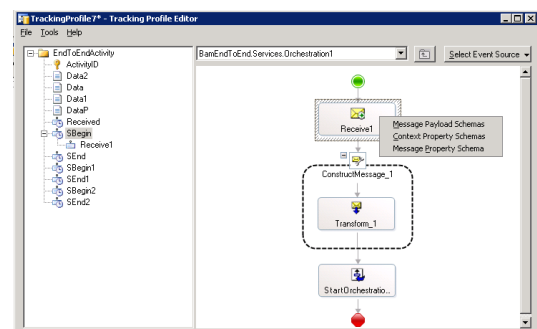
| Operation               |      | Year | TotalRequest | 76   | 16      | 23      | 13  | 128  |
|-------------------------|------|------|--------------|------|---------|---------|-----|------|
| PEMCancelDomainPT       |      |      | TotalRequest | 76   | 16      | 23      | 13  | 128  |
| PEMChangeStatusDomainPT |      |      | TotalRequest | 1413 | # 145 # | 32      | 195 | 1619 |
| PEMCreateDomainPT       |      |      | TotalRequest | 327  | 2       | 17      | 53  | 214  |
|                         | 2009 |      | TotalRequest | 1107 | 3       | 5       | 108 | 1223 |
|                         | 2010 |      | TotalRequest | 177  | 2       | 26      |     | 205  |
|                         | 2011 |      | TotalRequest | 1611 | 7       | 5       | 151 | 2041 |
| Grand Total             |      |      | TotalRequest | 3100 | 7       | # 312 # | 108 | 422  |



Estas ferramentas permitem libertar os programadores da preocupação, na altura do desenvolvimento, de qual a informação que pretendem recolher e da implementação da monitorização, uma vez que os analistas podem desenvolver o seu modelo com uma ferramenta extremamente familiar (Microsoft Excel) e o administrador poderá posteriormente ligar o modelo com os processos utilizando a ferramenta *Tracking Profile Editor*.

The screenshot shows the 'Tracking Profile Editor' with a data table titled 'TotalPedidos'. The table displays request data categorized by operation and year.

| Operation        | CodeEmis   | Destino_1 | Destino_2 | Destino_3 | Total | Grand |
|------------------|------------|-----------|-----------|-----------|-------|-------|
| Operacao_1       | CodeEmis_1 | 3         | 2         | 3         | 2     |       |
| Operacao_2       | CodeEmis_2 | 4         | 1         | 5         | 2     |       |
| Operacao_3       | CodeEmis_3 | 1         | 1         | 1         | 3     |       |
| Operacao_1 Total |            | 4         | 4         | 8         |       |       |
| Operacao_2 Total |            | 5         | 3         | 2         | 10    |       |
| Operacao_3 Total |            | 1         | 1         | 1         | 3     |       |
| Operacao_1 Total |            | 1         | 1         | 2         |       |       |
| Operacao_2 Total |            | 1         | 1         | 1         | 3     |       |
| Operacao_3 Total |            | 3         | 1         | 4         |       |       |
| Operacao_3 Total |            | 5         | 2         | 2         | 9     |       |
| Total Grand      |            | 19        | 11        | 5         | 35    |       |



O grande benefício do BAM é que proporciona a visibilidade sobre os processos de negócios. Este pequeno grande efeito, poderá ser o ideal para envolver os responsáveis de negócio na operação activa dos processos criando um *feedback loop* importantíssimo para a melhoria contínua dos projectos de automação (BPA).

Uma vez que a implementação do BAM é muitas vezes apenas a definição do modelo e toda a implementação é automatizada (geradores), o esforço

# Microsoft BizTalk Server aos olhos dos programadores

para incluir o BAM é muito simplificado para os programadores.

Mais informações: [Business Activity Monitoring](#)

## BizTalk WCF Service Publishing Wizard e BizTalk WCF Service Consuming Wizard

Mesmo quando for preciso expor um processo como um Web Service, existem *wizards* que nos ajudam a criar tanto os projectos WCF, como na instalação destes no servidor Web (IIS). O mesmo acontece no caso de querermos incluir nos nossos processos, chamadas a outros Web Services.

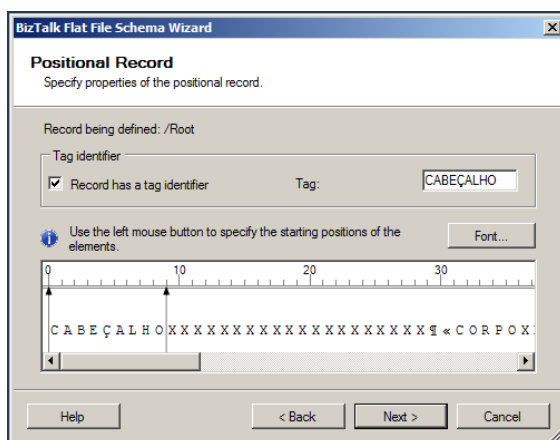
Estas ferramentas permitem aos programadores abstraírem-se de programação repetitiva, agilizando o processo de integração.

## BizTalk Flat File Schema Wizard

Uma dos padrões mais antigos para a troca de mensagens é a utilização de arquivos texto (*Flat Files*) como como: CSV ou TXT, muitos deles customizados à medida para os sistemas. Porém com a adopção do XML como formato de eleição na troca de mensagens, muitas vezes é necessário transformar arquivos texto em XML e vice-versa.

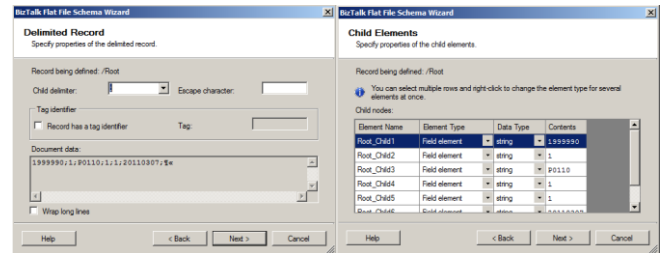
A ferramenta “*BizTalk Flat File Schema Wizard*” permite ao programador facilmente e de forma visual efectuar transformação de arquivos de texto posicionais:

```
CABEÇALHOXXXXXXXXXXXXXXXXXXXXX
CORPOXXXXXXXXXXXXXXXXXXXXXXXXXX
CORPOXXXXXXXXXXXXXXXXXXXXXXXXXX
RODAPÉXXXXXXXXXXXXXXXXXXXXXXXXX
```



Ou delimitado por símbolos:

```
1999990;1;P0110;1;1;20110307;
1999990;2;P0529;2;2;20110307;
1999990;3;P0530;3;3;20110307;
```



A definição das regras de “*parsing*” fica embebida no *schema* XSD, simplificando logo toda a reutilização destes esquemas nos diferentes pontos do processo. Em qualquer ponto o documento poderá ser novamente traduzido para *flat-file* pois a definição é declarativa e simétrica.

## Testes, debug e suporte ao Team Foundation Server

As ferramentas de desenvolvimento do BizTalk usam pastas, ficheiros e projectos do Visual Studio. Assim as equipas podem tirar partido de todo o ambiente ALM (*Application Lifecycle Management*) colaborando naturalmente com os demais programadores, gestores de projecto e *testers* que fazem parte da equipa.

Claro que dentro do VS vão encontrar suporte para *debug*, seja dos *orchestrations*, ou dos mapas (XSLT), ou componentes de *pipelines*. Bem como para a realização de Testes Unitários e muitas outras ferramentas do Visual Studio (da Microsoft ou de outros fornecedores).

## Conclusão

Conforme apresentado neste artigo, com o foco no desenvolvimento de soluções de integração empresarial, o BizTalk Server 2010 simplifica a integração de aplicações, independentemente dos seus interfaces se encontrarem dentro (EAI), nos seus parceiros (B2B) ou alojados na cloud.

Este tipo de *middleware* torna-se cada vez mais importante quando o número de *endpoints* cresce cada vez mais e o tipo de mensagens transaccionadas se torna cada vez mais crítico num mercado cada vez mais electrónico e automatizado.

## Autor



Escrito por Sandro Pereira

Actualmente Senior Software Developer na empresa DevScope. É Microsoft Most Valuable Professional (MVP) em Microsoft BizTalk. O seu principal foco de interesse são as tecnologias e plataformas de Integração (EAI): BizTalk e SOAP / XML / XSLT e Net, que utiliza desde 2002.

# Microsoft BizTalk Server aos olhos dos programadores

É um participante bastante activo nos fóruns da Microsoft (MSDN BizTalk Server Forums) e autor do blog <http://sandroaspbiztalkblog.wordpress.com> - Twitter: @sandro\_asp