

6 DE FEVEREIRO DE 2017



ARQUITETURA BPM

INTERFILE BPM

RODRIGO CARLSTROM

INTERFILE BPO

www.interfilebpo.com.br

Sumário

Objetivo.....	2
Visão Geral	2
Benefícios	2
Arquitetura.....	3
Estrutura de projeto.....	4
Contas de consumo.....	5

Objetivo

Este documento apresenta as soluções adotadas para atender aos requisitos técnicos além de facilitar e padronizar a codificação dos requisitos funcionais dos sistemas. A equipe técnica do projeto poderá, a partir deste documento, discutir e validar a arquitetura do sistema.

As soluções definidas aqui são propostas para servir como base no desenvolvimento das fases do projeto, em conjunto com os artefatos gerados durante as fases de levantamento e especificação.

Visão Geral

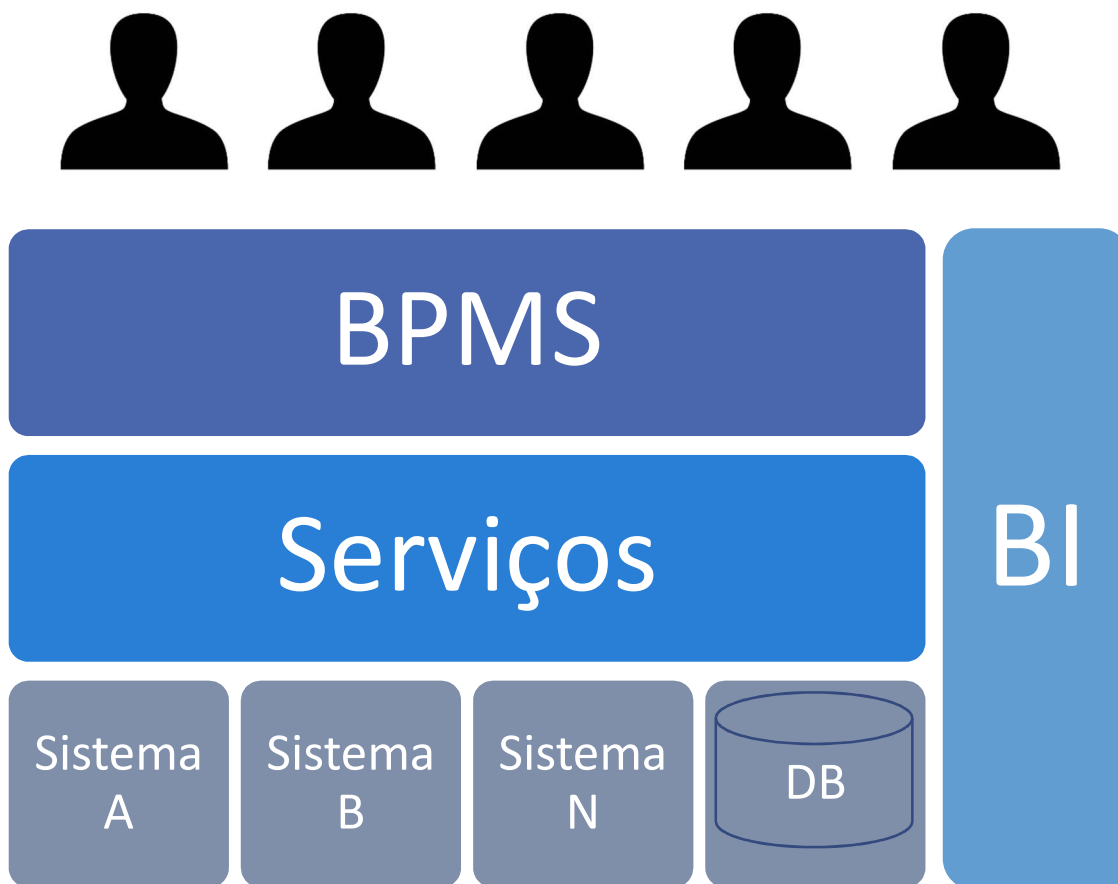
A arquitetura apresentada a seguir leva em consideração, prioritariamente, a utilização do BPMS, as integrações com outros sistemas e o desenvolvimento de projetos em paralelo. Além destes dois cenários, a solução adotada também pretende contemplar os requisitos de segurança, reutilização e desempenho.

Benefícios

Aqui estão alguns benefícios provenientes da arquitetura do projeto:

- Reutilização de componentes e funcionalidades;
- Alta coesão e atribuição de responsabilidades com base nas camadas lógicas (apresentação, negócio e acesso a dados);
- Facilidade na manutenção de eventuais problemas técnicos, devido à separação de camadas estruturais e funcionais (baixo acoplamento);
- Padronização de código;
- Facilidade para expor funcionalidades de negócio através de serviços.

Arquitetura



- **BPMS¹:** Ferramenta de modelagem, desenvolvimento, execução e monitoramento de processos de negócio, composta por:
 - **Modeler:** Modelador de processos em BPMN² e de regras de negócio em DMN³, utilizado para expressar graficamente os processos de negócios com as características técnicas e de automação que os negócios demandam. Deve ser modelado o processo de execução, ao invés do processo didático somente;
 - **Admin:** Interface de administração de usuários, grupos e autorizações em geral;
 - **Cockpit:** Interface de gestão operacional dos processos de negócios em execução (visão presente para ação imediata);
 - **Tasklist:** Interface de operação dos processos, onde os participantes podem iniciar novos processos e responder às atividades de processos em execução;
 - **REST⁴:** API de interface de acesso ao motor de execução de processos;
- **Serviços:** Camada de serviços para atendimento aos processos de negócio, dividida em:
 - **Java:** Implementação de funcionalidades em Java para persistência na base de dado (via JPA⁵), implementação de integrações com outros sistemas;
 - **SOA:** Implementação de barramento para acesso por outros sistemas, provendo interface de acesso. O SOA deve-se valer dos componentes java (jars) de forma a permitir o reaproveitamento;

- **Sistemas/DB:**
 - **Sistemas:** Disponibilidade tecnológica de apoio ao negócio, como software de prateleira ou de parceiros. Nesta arquitetura o usuário (participante do processo de negócio) não interage diretamente com os sistemas, mas sim por meio do BPMS, de forma a padronizar o *modus operandi* da operação e facilitar o treinamento;
 - **DB:** Base de dados de negócio, para acesso à camada de BI⁶ e para busca histórica. Deve ser acessível para leitura/escrita pelo JPA da camada de serviços e somente leitura pelo BI;
- **BI:** Portal de indicadores composto de:
 - **Dashboard:** Indicadores de processo (visão histórica para prospecção de um futuro melhor). Com base nas informações do BPMS. Acesso via REST;
 - **Negócios:** Indicadores de negócio (visão qualitativa e quantitativa). Com base nas informações da Base de Dados, com acesso a conexão somente leitura.

Estrutura de projeto

A estrutura de projeto foi pensada em atender as necessidades de reaproveitamento do de funcionalidade comuns e também a necessidade de atender as diferenças de negócios dos diversos cliente da Interfile. A estrutura está formatada como segue:

Tabela 1 - Estrutura de Projeto

Nível	Projeto/Subprojeto	Dependência	Tipo Maven
1.0	Interfile		POM
1.1	Interfile-bpm	Interfile-core	WAR
1.1	Interfile-core		JAR
2.0	Interfile-<seg>		POM
2.1	Interfile-<seg>-bpm	Interfile-<Seg>-core	WAR
2.1	Interfile-<Seg>-core	Interfile-core	JAR
3.0	Interfile-<seg>-<neg>		POM
3.1	Interfile-<seg>-<neg>-bpm	Interfile-<seg>-<neg>-core	WAR
3.1	Interfile-<seg>-<neg>-core	Interfile-<Seg>-core	JAR
4.0	Interfile-<seg>-<neg>-<cli>		POM
4.1	Interfile-<seg>-<neg>-<cli>-bpm	Interfile-<seg>-<neg>-<cli>-core	WAR
4.1	Interfile-<seg>-<neg>-<cli>-core	Interfile-<seg>-<neg>-core	JAR
NA	Interfile-soa		POM
NA	Interfile-soa-web (barramento)	Interfile-soa-core	WAR
NA	Interfile-soa-core	Interfile-core Interfile-<Seg>-core Interfile-<seg>-<neg>-core	JAR

¹ BPMS: Business Process Management Suite

² BPMN: Business Process Management Notation

³ DMN: Decision Manager Notation

⁴ REST: Representational State Transfer

⁵ JPA: Java Persistence API

⁶ BI: Business Intelligence

Interfile-<seg>-<neg>-<cli>-core

Onde:

- **<seg>**: Seguimento de negócio, exemplo: Saúde, financeiro, etc;
- **<neg>**: Negócio, exemplo: Reembolso Saúde, Compensação de Cheque, etc;
- **<cli>**: Nome do cliente.

Nos níveis superiores tende-se a implementar na camada CORE, micro serviços mais genéricos e reaproveitáveis, por exemplo: Uma consulta de CPF poderia ser implementada do *Interfile-core*, assim estaria disponível a todos os projetos de nível inferior. Uma consulta de CRM poderia ser implementada no *Interfile-health-core* ficando disponível aos projetos de saúde de nível inferior.

Nos níveis inferiores tende-se a implementar na camada BPM, processos específicos às necessidades dos clientes, porém valendo-se da flexibilidade que um BPMS pode oferecer.

Contas de consumo

No BPMS o atributo *tenantId*, permite que processos, usuário e grupos com *tenantId* em comum, sejam somente visíveis entre si, assim uma mesma instalação pode atender a diversas operações. Cada projeto da **Tabela 1 - Estrutura de Projeto** tem seu *tenantId*. Esta forma de trabalho fortalece o reaproveitamento respeitando a necessidade de singularidade de cada cliente bem como a confidencialidade de informações. Além de prover uma melhor organização de artefatos de desenvolvimento.

Convergente a esta mesma linha de pensamento, o barramento de serviços deve ser estruturado da mesma forma. No barramento então haverá uma credencial de produto (um código que será criado pelo próprio barramento de forma administrativa). Este código será configurado como o *tenantId* para o BPMS, mantendo toda a rastreabilidade de consumo dos serviços pelos processos. Os endereçamentos REST na construção dos serviços devem obedecer a mesma lógica, por exemplo:

```
GET /cnpj/03778032000161/status
```

```
GET /health/crm/12459751/status
```

Neste barramento haverá um gateway que tem a funcionalidade de, além de delegar para o serviço requisitado, armazenar para consulta futura as requisições com as respectivas credenciais de produtos (*tenantId*) e seus resultados. Com isto será possível gerar indicadores e também ratear o custo de TI com as diversas operações.