

RAFAEL PORCIDONIO FERNANDES ESQUIÇATO

**ANÁLISE E PROJETO DA ARQUITETURA DE UM PORTAL
CORPORATIVO PARA UMA COMPANHIA DE SEGUROS**

Monografia apresentada ao PECE – Programa de Educação Continuada em Engenharia da Escola Politécnica da Universidade de São Paulo como parte dos requisitos para conclusão do curso de MBA em Tecnologia de Software.

São Paulo
2013

RAFAEL PORCIDONIO FERNANDES ESQUIÇATO

**ANÁLISE E PROJETO DA ARQUITETURA DE UM PORTAL
CORPORATIVO PARA UMA COMPANHIA DE SEGUROS**

Monografia apresentada ao PECE – Programa de Educação Continuada em Engenharia da Escola Politécnica da Universidade de São Paulo como parte dos requisitos para a conclusão do curso de MBA em Tecnologia de Software.

Área de Concentração: Tecnologia de Software

Orientador: Prof. Dr. Maria Alice Grigas Varella Ferreira

São Paulo
2013

FICHA CATALOGRÁFICA

Esquiçato, Rafael Porcidonio Fernandes

Análise e projeto da arquitetura de um portal corporativo para uma companhia de seguros / Rafael Porcidonio Fernandes Esquiçato. -- São Paulo, 2013.

Monografia – Escola Politécnica da Universidade de São Paulo. PECE, Programa de Educação Continuada em Engenharia.

1. Portais corporativos 2. Arquitetura de software 3. Internet
4. Seguros I. Universidade de São Paulo.
Escola Politécnica. PECE, Programa de Educação Continuada em Engenharia.

DEDICATÓRIA

Dedico este trabalho aos meus pais Laércio e Isabel que me ensinaram o valor da educação e do esforço e à minha namorada Ana Cláudia que me apoiou e compreendeu meus momentos de ausência pra finalizar esse trabalho.

AGRADECIMENTOS

À professora Maria Alice Grigas Ferreira, que aceitou me orientar durante o desenvolvimento desse trabalho.

À Universidade de São Paulo – USP que me forneceu um ambiente adequado para pesquisar e desenvolver meu trabalho, através de seus professores e parcerias.

À equipe da Tecnologia Única que me deu apoio e suporte durante todo o desenvolvimento do trabalho me ajudando a entender o mercado de seguros.

À todos os meus amigos que me apoiaram e compreenderam os momentos em que estive ocupado. Em especial ao pessoal da Escola Técnica Fernando de Prestes e da Faculdade de Tecnologia de Sorocaba que representam grandes amizades muito presentes até hoje.

À toda família Porcidonio que me apoiou e me deu muita alegria pra continuar trabalhando.

RESUMO

Com o nível crescente de informatização das organizações, vários sistemas são desenvolvidos separadamente para dar vazão às demandas das instituições. Por esse motivo, ilhas de conhecimento são geradas dentro das organizações. Cada sistema possui o seu modelo de dados, sua forma de operação e, muitas vezes, perdem-se muitas oportunidades de criação de novas funcionalidades devido à ausência de um centro integrado de informações. A ideia desse trabalho é apresentar e analisar uma solução para esse problema: um portal corporativo (EIP). Ele pode reunir, em um único ponto de acesso, informações e funcionalidades dos diversos sistemas da companhia. Esse ponto de acesso pode ser usado por usuários internos - colaboradores da organização - e externos - fornecedores e clientes. Nesse trabalho discute-se a arquitetura de um portal corporativo, analisando-se as diversas visões que a compõe. As visões de negócio, de tecnologia e de organização são apresentadas e discutidas. Os recursos tecnológicos envolvidos bem como as alternativas de projeto complementam a discussão. Ao final do trabalho, um projeto prático é apresentado com base no caso real de uma grande seguradora.

ABSTRACT

With the increasing automatization level of organizations, many systems are developed separately to give vent to the institutions demands. Therefore, knowledge islands are generated within organizations. Each system has its own data model, its mode of operation and often many opportunities are lost to create new features due to the absence of an integrated information center. We present and analyze a solution to this problem: an enterprise information portal (EIP). It can gather, in a single point of access, information and functionality from the various systems of the company. This access point can be used by internal users - organization employees - and external - suppliers and customers. We discuss the architecture of a corporate portal, analyzing the various views that compose it. Business, technology and organization views are presented and discussed. Technological resources involved and the design alternatives complement the discussion. At the end of the paper, a practical project is presented based on the real case of a large insurance company.

LISTA DE ILUSTRAÇÕES

	Pág.
Figura 1. Cadeia de conhecimento, extraída de WU e WANG (2009).....	23
Figura 2. Arquitetura de uma solução de KM baseada em portal, extraída de WU e WANG (2009).....	25
Figura 3. Hierarquia dos atributos de qualidade externa / interna (extraída de NBR ISO/IEC 9126-1).....	27
Figura 4. Pirâmide de formas de autenticação, extraída de Polgar, Bram e Polgar (2006). .	30
Figura 5. Diagrama de seqüência com fluxo de SSO por tíquete.....	32
Figura 6. Posicionamento da camada de integração.....	40
Figura 7. Integração via portal, extraída de Trowbridge et al. (2004).	46
Figura 8. Arquitetura da camada de integração composta pelas três alternativas elencadas.	48
Figura 9. Modelo de projeto do portal	76

LISTA DE QUADROS

	Pág.
Quadro 1. Requisitos funcionais do portal corporativo.	62
Quadro 2. Requisitos não funcionais do portal.....	63
Quadro 3. Sistemas de back-office envolvidos.....	64
Quadro 4. Funcionalidades no portal e soluções de implementação.....	78
Quadro 5. Serviços a serem implementados nos sistemas envolvidos com o portal.....	78
Quadro 6. Classificação dos componentes de acordo com a área de negócio.....	79

LISTA DE ABREVIATURAS E SIGLAS

ACM	<i>Association for Computing Machinery</i>
B2B	<i>Business-to-Business</i>
B2C	<i>Business-to-Customer</i>
B2E	<i>Business-to-Employee</i>
BI	<i>Business Intelligence</i>
COM	<i>Component Object Model</i>
CORBA	<i>Common Object Request Broker Architecture</i>
CSS	<i>Cascading Style Sheets</i>
CMS	<i>Content Management System</i>
CRM	<i>Customer Relationship Management</i>
DDD	<i>Domain Driven Development</i>
EAI	<i>Enterprise Application Integration</i>
EIP	<i>Enterprise Information Portal</i>
ERP	<i>Enterprise Resource Planning</i>
FTP	<i>File Transport Protocol</i>
HTML	<i>Hypertext Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
KDS	<i>Knowledge Discover Server</i>
KM	<i>Knowledge Management</i>
MSDN	<i>Microsoft Developer Network</i>
LTPA	<i>Lightweight Third Party Authentication</i>
PIN	<i>Personal identification number</i>
RH	<i>Recursos Humanos</i>
SCM	<i>Supply Chain Management</i>
SOA	<i>Service-Oriented Architecture</i>
SOAP	<i>Simple Object Access Protocol</i>
SSO	<i>Single Sign-On</i>
URL	<i>Uniform Resource Location</i>
W3C	<i>World Wide Web Consortium</i>
XML	<i>EXtensible Markup Language</i>

SUMÁRIO

	Pág.
1. INTRODUÇÃO	13
1.1. Motivações	13
1.2. Objetivo	14
1.3. Metodologia	14
1.4. Estrutura de trabalho	15
2. REVISÃO BIBLIOGRÁFICA	17
3. ESTADO DA ARTE	20
3.1. Visão de negócio	21
3.1.1. <i>Business-to-Business (B2B)</i>	21
3.1.2. <i>Business-to-Customer (B2C)</i>	22
3.1.3. <i>Business-to-Employee (B2E)</i>	22
3.2. Gestão de conhecimento	23
3.3. Qualidade de Software	26
3.4. Visão tecnológica	29
3.4.1. Autenticação e <i>Single Sign-On (SSO)</i>	29
3.4.2. Ferramentas de Gerenciamento de Conteúdo	34
3.4.3. Pesquisa de Documentos	37
3.5. Integração de aplicações	37
3.5.1. Camada de integração	40
3.5.2. Métodos de conexão	49
3.6. Considerações finais do capítulo	60
4. DESENVOLVIMENTO	61
4.1. Projeto de um portal corporativo	61
4.1.1. Requisitos funcionais	62
4.1.2. Requisitos não funcionais	63
4.1.3. Sistemas de <i>back-office</i>	64
4.1.4. Estrutura básica	65
4.1.5. Funcionalidades	69
4.1.6. Modelo de arquitetura	75
4.1.7. Resumo das integrações	77
4.1.8. Classificação dos componentes	78
4.2. Visão organizacional	79
4.3. Considerações finais do capítulo	81
5. ANÁLISE DE RESULTADOS	82
5.1. Benefícios esperados	82

5.2.	Dificuldades esperadas.....	82
6.	CONSIDERAÇÕES FINAIS.....	84
6.1.	Conclusão.....	84
6.2.	Contribuição.....	84
6.3.	Trabalhos futuros.....	84

1. INTRODUÇÃO

Este capítulo apresenta as motivações, o objetivo, as justificativas e a metodologia adotada para o trabalho. No final do capítulo, a estrutura do trabalho é apresentada junto a uma descrição de cada capítulo que a compõe.

1.1. Motivações

O ambiente corporativo é formado por diversas aplicações que são utilizadas por uma gama variada de usuários, gerando e consumindo informação através delas. Num ambiente como esse, ilhas de conhecimento são geradas (WU; WANG, 2009). Dessa forma, alguma iniciativa se faz necessária para que o conhecimento gerado nessas ilhas possa chegar aos usuários que precisam dele, quer sejam de dentro da empresa ou de fora dela.

Um portal corporativo (um EIP ou *Enterprise Information Portal*) pode ser essa iniciativa, utilizando recursos tecnológicos, como páginas e serviços Web. Através de um portal corporativo que implemente a integração das informações, as ilhas de conhecimento podem ser eliminadas, aumentando a competitividade da empresa (WU; WANG, 2009). Com as informações circulando mais facilmente entre os setores da organização, os processos podem ser otimizados causando uma diminuição do custo e assim gerando esse aumento da competitividade da empresa no mercado.

Nos estágios iniciais da internet, os *sites* possuíam poucos recursos, como páginas estáticas para exibir informações aos usuários. Com o passar do tempo, eles se tornaram mais complexos, exibindo informações dinâmicas através de linguagens de *script* como ASP, PHP e outras.

Segundo Serhani, Al-Qirim e Benhareef (2010), nos dias de hoje, pode-se definir um portal como um conjunto de aplicações dinâmicas, exibindo diversos conteúdos através de um único ponto de acesso.

Para atingir os objetivos mencionados, um portal corporativo deve consultar informações de uma série de origens disponíveis no ambiente da empresa, como ERPs (*Enterprise Resource Planning*), sistemas de *supply chain*, sistemas

financeiros, sistemas de Recursos Humanos (RH), além de sistemas de BI (*Business Intelligence*). Segundo Tang e Chen (2007), um sistema de portal como esse deve coletar, processar, analisar e disponibilizar as informações originárias das fontes citadas acima.

Nesse sentido, a conhecida disciplina de integração de aplicações (*Enterprise Application Integration* - EAI) visa fornecer os métodos adequados para essa troca de dados. O objetivo final da EAI nesse contexto, é utilizar o portal como um centro de dados, fornecendo uma plataforma para novos desenvolvimentos (TANG; CHEN, 2007).

A motivação para tratar desse assunto surge dos fatores citados até aqui. Encontrar uma arquitetura que permita ao portal desempenhar esse papel de centro de conhecimento é algo desafiador e que merece uma série de considerações.

1.2. Objetivo

O objetivo do trabalho é apresentar as práticas realizadas nas fases de análise e projeto do portal corporativo de uma seguradora, sendo que o resultado final esperado é uma especificação da arquitetura desse portal.

Durante o desenvolvimento da monografia, diversos aspectos técnicos e de negócio pertinentes às duas fases são apresentados e discutidos, passando por fatores como acesso a dados, integração com aplicações, interface gráfica amigável para usuários não especialistas e funcionalidades em comum a várias categorias de usuários.

Por outro lado, esse trabalho não visou a implementação final do projeto, uma vez que não haveria tempo hábil – nem equipe - para tal. Por essa razão não se propõe aqui examinar aspectos relativos à avaliação quantitativa do mesmo.

1.3. Metodologia

O trabalho começou com a definição de objetivos baseados na literatura disponível sobre o assunto e na contribuição que seria atingida na aplicação dos conceitos num caso real. O material encontrado nas pesquisas feitas está disponível nas

referências do trabalho, além de ser apresentado, no capítulo 2 (Revisão Bibliográfica).

O principal meio de pesquisa utilizado foi a internet, em portais de origem confiável e verificada, como os portais do IEEE – *Institute of Electrical and Electronics Engineers* – e da ACM – *Association for Computing Machinery*. Artigos extraídos desses portais serviram como referência para grande parte do trabalho. Livros sobre o assunto também serviram como base de conhecimento. Além disso, cabe destacar a importância do portal .periódicos. da CAPES¹ acessível em qualquer ponto do campus da USP.

Os conceitos adquiridos foram utilizados como base para apresentar o estado da arte do projeto de portais corporativos. Temas como infraestrutura, funcionalidades comuns, serviços, integração de aplicações e gestão de conhecimento são abordados ao longo do trabalho com o objetivo de analisar as alternativas dentro de cada ponto e a contribuição que cada um pode trazer à solução final.

Ao final do trabalho, o conhecimento apresentado é aplicado a um caso real de uma grande seguradora do mercado brasileiro no ramo de vida. O objetivo é definir o portal dessa organização, através de sua arquitetura - visão tecnológica e de negócio - e os recursos que ele deve possuir.

1.4. Estrutura de trabalho

No primeiro capítulo do trabalho (INTRODUÇÃO), as motivações, o objetivo, as justificativas, a metodologia e a estrutura do trabalho são apresentados.

No segundo capítulo do trabalho (REVISÃO BIBLIOGRÁFICA), os resultados da pesquisa realizada são apresentados, elencando os principais artigos e livros consultados e analisando a contribuição de cada um para o trabalho final.

¹ <http://www.periodicos.capes.gov.br.ez67.periodicos.capes.gov.br/>

No terceiro capítulo (ESTADO DA ARTE), o tema é desenvolvido elencando diversos fatores que fazem parte da arquitetura de um portal corporativo e que devem ser estudados. Primeiramente, discutem-se as grandes áreas de funcionalidades de negócios que o portal envolve. Em seguida trata-se do problema da gestão de conhecimento dentro das organizações. Considerando-se que o portal está envolvido com diversas funcionalidades diferentes referentes à empresa, a seguir discute-se os problemas referentes à integração das funcionalidades e serviços através dele.

No quarto capítulo (DESENVOLVIMENTO) aplicam-se os conceitos examinados no capítulo 3 no desenvolvimento de um portal corporativos para um negócio do ramo de seguros. O resultado esperado desse capítulo é a definição da arquitetura do portal proposto utilizando os conceitos elencados.

No quinto capítulo (ANÁLISE DOS RESULTADOS), uma análise do impacto da implantação do portal proposto no capítulo de desenvolvimento é apresentada.

No sexto capítulo (CONSIDERAÇÕES FINAIS), o trabalho encerra-se apresentando as contribuições acadêmicas do mesmo, bem como alguns pontos não abordados e que podem servir em trabalhos futuros.

2. REVISÃO BIBLIOGRÁFICA

O livro de Polgar, Bram e Polgar (2006) retrata conceitos fundamentais do funcionamento de um portal. O livro é bastante abrangente e fala sobre aspectos de negócio, como definição do público-alvo e aspectos mais técnicos como protocolos e ferramentas de desenvolvimento. No trabalho, teve grande contribuição na discussão sobre formas de autenticação e *Single Sign-On*, trazendo alguns métodos importantes. Também trouxe grande contribuição nas discussões sobre os públicos-alvo de um portal corporativo, assim como o posterior mapeamento das funcionalidades dentro dos públicos.

O artigo de Wu e Wang (2009) retrata os problemas da gestão de conhecimento nas empresas e a solução que pode ser obtida através de um portal (EIP) que agregue dados de vários sistemas, gerando indicadores comuns a toda a comunidade, além de elencar os recursos importantes que um portal, do ponto de vista da gestão interna de conhecimento, deveria ter como: espaços virtuais para reuniões, bibliotecas de documentos e outras coisas. No trabalho, teve contribuição nas discussões sobre gestão do conhecimento.

A norma NBR ISO/IEC 9126-1 (ABNT, 2003) define os atributos de qualidade de software. No trabalho, a contribuição da norma foi no sentido de elencar quais desses atributos são mais prioritários em uma solução de portal corporativo.

O livro de Watt (2002) aborda o projeto e construção de soluções de negócio através da Web. O trabalho aborda desde o gerenciamento do projeto até recursos técnicos. A contribuição para o trabalho foi a discussão sobre ferramentas de gestão de conteúdo.

O artigo de Chieu, Nguyen e Zeng (2007) apresenta um método de pesquisa de documentos privados baseado em anotações XML. A contribuição para o trabalho foi conceituar alguns pontos na discussão sobre gestão de documentos.

Os trabalhos de Hohpe e Woolf (2003) e Trowbridge e seus colegas (2004) tratam de padrões para a integração de aplicações. Nos dois trabalhos, diversas formas são apresentadas e seus benefícios e desvantagens são elencados. É importante

ressaltar que os padrões não sofrem interferência da tecnologia usada na implementação dos mesmos, já que são padrões de projeto. A contribuição para o trabalho foi discutir esse catálogo de padrões e aplicá-lo nas situações impostas pelo projeto do portal.

O trabalho de Tang e Chen (2007) é estudo de caso da implementação de um portal corporativo em uma indústria chinesa. O artigo define conceitos importantes sobre portais e também relaciona a solução padrões de integração de sistemas. O artigo também trata das funcionalidades comuns a um portal e também de questões de layout. A contribuição ao trabalho foi com a definição de alguns conceitos sobre portais.

O artigo de Tiwari e Joshi (2009) apresenta uma interessante forma de se implementar o processo de *Single Sign-On* entre duas aplicações sem a utilização de ferramentas comerciais baseado em criptografia e temporização. A contribuição do trabalho foi trazer mais uma forma de se implementar o processo de Single Sign-On.

O artigo de Wan et al. (2005) aborda ferramentas de gestão de conteúdo e conhecimento na Web. Tópicos como o processo de seleção e funcionalidades desejáveis são discutidos. A contribuição para o trabalho foi trazer uma nova perspectiva para a discussão sobre gestão de conteúdo.

O dicionário do Gartner Group (Gartner, 2012) define diversos termos envolvendo Tecnologia da Informação. Para o trabalho teve grande contribuição na definição dos termos elencados no glossário do trabalho.

O site educacional da W3C, o W3C Schools (W3C, 2012), contém diversos minicursos sobre tecnologias para internet, como HTML, *JavaScript*, CSS, SOAP (*Simple Object Access Protocol*) e outras coisas. Para o trabalho, contribuiu na discussão sobre integração via serviço, definindo o protocolo SOAP utilizado na criação de *WebServices*.

Os sites sobre tecnologias Microsoft, como o MSDN e o TechNet (MICROSOFT, 2012b), dispõem de um grande acervo de artigo técnicos sobre ferramentas da

empresa. Para o trabalho, a contribuição foi nas discussões sobre funcionalidades comuns de um portal corporativo já que a Microsoft dispõe do SharePoint que é uma ferramenta nesse sentido. A solução de *Single Sign-On* da Microsoft contribuiu na discussão sobre esse recurso comum dentro de um portal corporativo.

3. ESTADO DA ARTE

Nesse capítulo, o panorama conceitual do trabalho é apresentado. A experiência profissional do autor e as pesquisas feitas servem como base para apresentar o conteúdo.

Na primeira parte do capítulo, são apresentados os diversos conceitos que envolvem a criação e a manutenção de um portal corporativo. O objetivo é dissertar sobre temas como B2B (*business-to-business*), B2C (*business-to-costumer*), B2E (*business-to-employee*) e demonstrar como eles compõem as grandes áreas de funcionalidades de negócio que um portal corporativo pode oferecer.

Na segunda parte, a criação de portais corporativos é apresentada como uma solução para os problemas de gestão de conhecimento (KM – *Knowlegde Management*) dentro das organizações. Através da literatura disponível, conceitos de gestão de conhecimento serão apresentados, assim como várias formas de implementá-los no portal de uma organização, servindo como uma base sólida para se consultar manuais de operação, perguntas frequentes, *templates* de documentos e outros itens de conhecimento.

Na terceira parte, são discutidas funcionalidades comuns oriundas de recursos tecnológicos como CMS (*Content Manager System*) e *Single Sign-On*. Esses recursos formam a infraestrutura para as funcionalidades de negócio a serem implementadas.

Na quarta parte do capítulo, serão apresentadas as diversas formas que os portais têm de trocar dados com outros sistemas. A literatura disponível sobre integração de aplicações servirá como base para essa discussão. Como já foi dito, um portal, para cumprir seu objetivo, deve consumir informações dos sistemas de *back-office* da organização (sistemas ERP, sistemas de SCM, sistemas de RH e outros). Nessa parte do trabalho, serão analisadas as alternativas para acessar dados de diversas formas.

3.1. Visão de negócio

Como definiram Polgar, Bram e Polgar (2006), do ponto de vista do cliente que o acessa, um portal é uma aplicação *Web* integrada e personalizada que provê, em um único acesso, uma grande variedade de conteúdo agregado (como dados e serviços).

Um portal corporativo surge da necessidade que a maioria das organizações têm de integrar o conhecimento gerado em diversos setores da companhia em um único ponto de acesso (WU; WANG, 2009). Com a implementação de um portal, clientes, parceiros e empregados podem compartilhar conhecimento, acelerando a execução dos processos empresariais.

Para atingir esse objetivo, os arquitetos e projetistas do portal precisam, como em qualquer outro sistema de informação, conhecer bem o negócio da companhia e quais serviços podem enriquecer o negócio. Segundo Polgar, Bram e Polgar (2006), desse ponto de vista, existem três grandes áreas de funcionalidades que podem ser implementadas em portal corporativo: *Business-to-Business (B2B)*, *Business-to-Customer (B2C)* e *Business-to-Employee (B2E)*.

3.1.1. ***Business-to-Business (B2B)***

São funcionalidades desenvolvidas para facilitar a integração da empresa com outras companhias (parceiros comerciais, fornecedores). Sistemas de SCM (*Supply Chain Management*) podem se beneficiar muito de funcionalidades e serviços como esse. Fornecedores e revendedores podem acessar funcionalidades tais como visualizar, fazer e acompanhar pedidos, agendar entregas, entre outras coisas. No contexto de uma seguradora, exemplos de serviços de B2B podem ser:

- Um serviço de cotação para que os corretores façam e acompanhem cotações de novos seguros;
- Um serviço para abertura de sinistros, onde os corretores possam abrir novos processos de sinistros na seguradora para seus clientes.

Com serviços como os citados, as empresas facilitam a relação com parceiros, agilizando a troca de informações por meio eletrônico. Por exemplo, quando um corretor realiza uma cotação e o cliente a aceita, ela já está registrada na seguradora e já pode ser aprovada, gerando uma nova apólice. Não há necessidade de entrar em contato com a seguradora através de uma central de atendimento ou de outro meio.

3.1.2. *Business-to-Customer (B2C)*

São funcionalidades desenvolvidas para os clientes da companhia. Através de serviços de B2C, clientes têm acesso a uma gama de conteúdos relativos aos seus negócios com a companhia, como manuais de produtos e listas de preços. Associados a ferramentas de CRM (*Customer Relationship Management*), é possível oferecer um acesso personalizado aos clientes com as informações necessárias, de acordo com os seus últimos negócios e interesses. No contexto de uma seguradora, serviços de B2C podem ser:

- Consulta às informações da apólice de um cliente: número, coberturas, vigência etc.;
- Consulta de parcelas vencidas, onde o próprio cliente pode acessar e emitir uma segunda via do boleto de pagamento.

Serviços como os citados acima, podem ajudar a melhorar a relação com o cliente, oferecendo serviços de autoatendimento sem a necessidade de ligar para um centro de atendimento.

3.1.3. *Business-to-Employee (B2E)*

São funcionalidades desenvolvidas para os empregados da companhia. Segundo Polgar, Bram e Polgar (2006), são uma forma de agregar e disseminar conteúdo corporativo para todos os funcionários da organização. Existem basicamente dois tipos de portal B2E:

- **Portais do funcionário** fornecem acesso geral às informações relevantes a todos os funcionários da organização, como notícias da companhia, aplicações

de RH, busca de documentos corporativos e outros tipos de conteúdos aplicáveis a grandes grupos de colaboradores. Esse tipo de portal também fornece funcionalidades de autoatendimento, onde os colaboradores podem se inscrever em programas de treinamento e benefícios ou ainda atualizar informações de contato;

- **Páginas profissionais** fornecem suporte a processos de negócio por meio de páginas do portal. Em contraste com os recursos do item anterior, elas são, geralmente, destinadas a apenas alguns papéis dentro da organização (como profissionais de TI, vendas ou RH). Através de portais como esse, os profissionais podem realizar tarefas de negócio, já que essas funcionalidades geralmente se integram com sistemas de *back-office*.

3.2. Gestão de conhecimento

Segundo WU e WANG (2009), nos dias atuais, a medida do valor de uma empresa foi alterada. Antes, a medida era feita com base na quantidade de prédios e equipamentos que uma empresa possuía; hoje, a propriedade intelectual, a confiança do mercado, a capacidade e a habilidade dos seus colaboradores também é levada em conta. Diante disso, é importante definir políticas e utilizar ferramentas para identificar e reter esse conhecimento.

A disciplina de gestão de conhecimento – *Knowledge Management* – tem basicamente esse objetivo e pode ser definida como o processo de identificar, adquirir, decompor, armazenar, transmitir, compartilhar e apreciar o conhecimento. A seguir, apresenta-se na figura 1 essa cadeia, onde cada bloco destaca cada uma das várias etapas mencionadas.

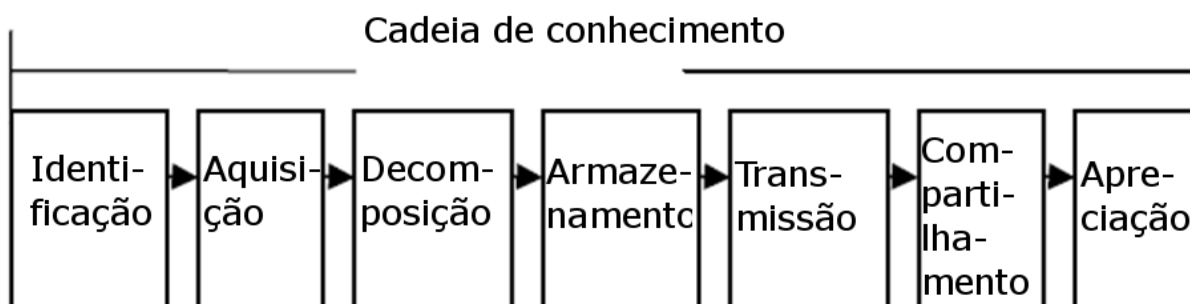


Figura 1. Cadeia de conhecimento, extraída de WU e WANG (2009)

O principal problema que uma solução de gestão de conhecimento deve solucionar dentro da organização é como criar uma estratégia que implemente toda essa cadeia e, ainda, fazê-lo de modo eficaz.

Aplicações de gestão de conhecimento devem, não apenas extrair dados das informações massificadas, mas também integrar as informações de cada indivíduo à da organização. A ideia é definir uma estratégia em que cada colaborador possa contribuir para o conhecimento da organização e não apenas consultá-lo. Assim, a solução escolhida deve permitir que cada colaborador possa contribuir com a sua experiência e conhecimento.

Portais corporativos são boas soluções para problemas de compartilhamento de conhecimento dentro da organização. Através de uma solução de portal, é possível criar uma plataforma de trabalho colaborativa, onde todos os envolvidos podem acessar informações sobre os projetos da companhia.

A plataforma colaborativa deve ser diferenciada da solução de e-mail corporativo. Em uma organização que não possui nenhuma solução que forneça essa plataforma, tudo tende a ser compartilhado por e-mail, o que traz alguns problemas:

- Acesso às informações: nem sempre a informação chega a quem deveria chegar, pois se acaba dependendo de quem gerou a informação selecionar para quem deseja enviar; nessa seleção, muitas vezes, a escolha de destinatários não é totalmente satisfatória;
- Confidencialidade: documentos com informações sensíveis trafegam livremente de um destinatário para outro sem haver um controlador global de acesso à informação;
- Tráfego na rede: como a única via de transmissão de informações e documentos é por e-mail, o volume de dados aumenta, já que um mesmo documento é enviado várias vezes para vários destinatários diferentes.

A solução de gestão de conhecimento baseada em portal corporativo cria a citada plataforma colaborativa, onde todos na organização podem acessar informações

pertinentes ao seu trabalho, como manuais de operação, documentos diversos e outras coisas. O portal, do ponto de vista do usuário, é enxergado como um escritório virtual, onde ele pode acessar de forma simples as notícias do dia, o conteúdo necessário para realizar suas tarefas e entrar em contato com outros funcionários da organização, facilitando a troca de informações, além de encontrar facilmente um especialista dentro da organização para resolver um problema que esteja tendo no momento.

A arquitetura de uma solução de KM baseada em portal é apresentada na Figura 2. Na arquitetura proposta, os componentes Área pessoal e Área comum formam o escritório virtual, onde o colaborador pode acessar todo o conteúdo destinado às suas tarefas. O portal corporativo é a infraestrutura do portal. O componente KDS (*Knowledge Discover Server*) é um servidor de aplicação baseado na Web e que é responsável por gerenciar os recursos de conhecimento disponíveis no portal, como gestão de documentos, indexação e recuperação de documentos, mineração de texto e *hyperlinks* com sistemas especialistas em um domínio específico. A interface é o componente responsável pela integração do portal corporativo e o KDS.

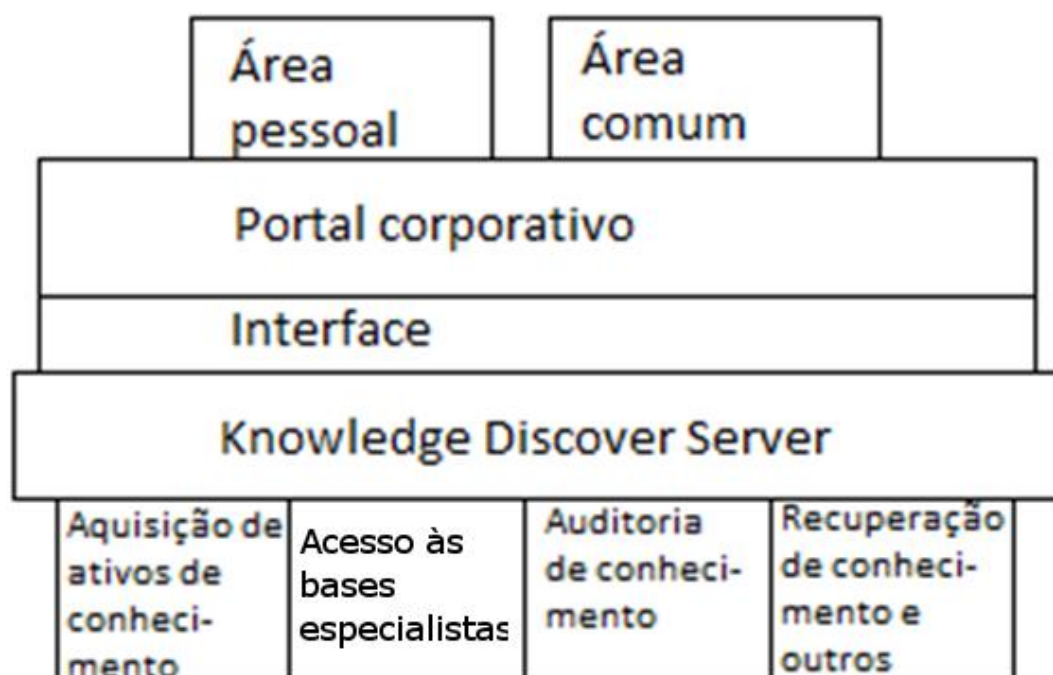


Figura 2. Arquitetura de uma solução de KM baseada em portal, extraída de WU e WANG (2009)

3.3. Qualidade de Software

Antes de discutir as questões técnicas envolvidas no desenvolvimento da descrição arquitetural de um portal corporativo, é importante definir alguns conceitos relacionados à qualidade de software. O objetivo da discussão é definir os atributos de qualidade mais relevantes ao contexto de portal corporativo seguindo a norma NBR ISO/IEC 9126-1, Engenharia de Software – Qualidade de Produto, Parte 1. – Modelo de Qualidade (ABNT, 2003).

A norma em questão especifica um modelo de qualidade externa e interna para o produto de software. Quanto à qualidade externa, consideram-se os atributos perceptíveis de fora da aplicação, enquanto a mesma é executada. Como a norma diz, esses atributos são geralmente avaliados num ambiente simulado, usando métricas externas.

Quanto à qualidade interna, consideram-se os atributos perceptíveis internamente. Os atributos em questão são geralmente medidos com base em modelos estáticos da aplicação, como código fonte e diagramas. As pontuações atingidas na análise desses atributos podem servir como base para validação das etapas intermediárias do desenvolvimento.

A figura 3 apresenta a hierarquia dos atributos de qualidade interna e externa definidos na norma. Cada uma das seis características principais é dividida em subcaracterísticas menores. Cada característica possui métricas próprias para avaliação e a soma dos atributos de qualidade obtidos para cada uma das características é definida como a qualidade do produto de software.

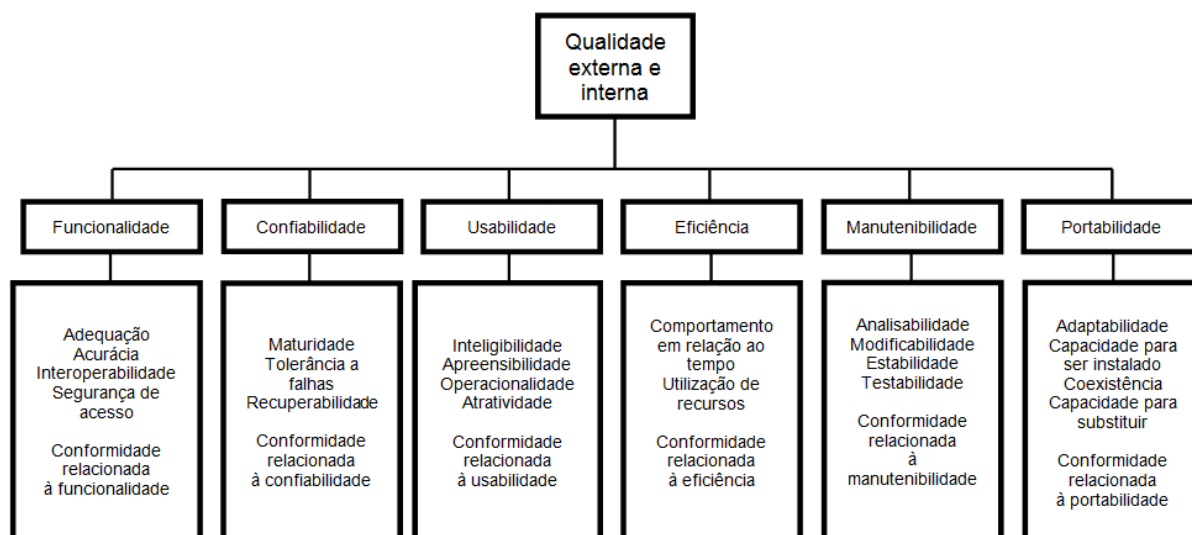


Figura 3. Hierarquia dos atributos de qualidade externa / interna (extraída de NBR ISO/IEC 9126-1).

Com relação aos portais corporativos, as seguintes subcaracterísticas têm maior relevância e precisam ser contempladas de alguma forma no projeto da solução:

- **Funcionalidade / Segurança de acesso:** capacidade do produto de software de proteger as informações e dados, de forma que sistemas não autorizados não possam lê-los nem modificá-los e que não seja negado o direito de leitura e modificação a usuários ou sistemas autorizados. Um portal expõe informações bastante sensíveis, como faturamento e tamanho de conta dos clientes da organização. Esses dados devem ser protegidos de acesso indevidos, seja de usuários desconhecidos ou mesmo de usuários conhecidos (por mais que um cliente tenha acesso ao portal, ele não deve ter acesso a dados de outros clientes, por exemplo).
- **Funcionalidade / Interoperabilidade:** capacidade do produto de software de interagir com um ou mais sistemas especificados. Como já foi dito, para um

portal corporativo atingir o objetivo de ser um ponto único de acessos para várias aplicações da organização, ele precisa se conectar à diversos sistemas para buscar informações. A capacidade da arquitetura descrita de tornar essas conexões ativas de forma simples é um atributo imprescindível.

- **Confiabilidade / Tolerância a falhas:** capacidade do produto de software de manter um nível de desempenho especificado em casos de defeitos no software ou de violação especificada. A dependência que um portal corporativo tem dos outros sistemas da companhia, dos quais consome ou para os quais gera informação, pode se tornar um problema se os mesmos forem instáveis. Nesse sentido, mecanismos de recuperação desse tipo de situação, sem que as funcionalidades fiquem indisponíveis no portal, precisam ser analisados.
- **Usabilidade / Apreensibilidade:** capacidade do produto de software de possibilitar ao usuário aprender sua aplicação. A gama de usuários de um portal corporativo é grande já que envolve empregados, parceiros e clientes. A quantidade de pessoas que necessitam de acesso à aplicação pode tornar inviável a realização de treinamentos. Dessa forma, as interfaces gráficas que compõem o portal devem ser fáceis de entender e utilizar, facilitando a adoção por parte de novos usuários.
- **Eficiência / Utilização dos recursos:** capacidade do produto de software de usar tipos e quantidades apropriados de recursos, quando o software executa suas funções sob as condições estabelecidas. Um portal corporativo possui forte ligação com os outros sistemas da organização. Alguns desses sistemas podem ser cruciais para operação da empresa e o portal precisa consumir os serviços necessários de forma otimizada para não impor uma lentidão desnecessária aos usuários dessas aplicações.
- **Manutenibilidade / Modificabilidade:** capacidade do produto de software de permitir que uma modificação seja implementada. A grande abrangência de um portal corporativo no que diz respeito às áreas de negócio envolvidas faz com que as necessidades de mudança, com o surgimento de novos requisitos, aconteçam frequentemente.

- **Manutenibilidade / Testabilidade:** capacidade do produto de software de permitir que o software, quando modificado, seja validado. A grande quantidade de integrações de um portal com outras aplicações pode torná-lo bastante suscetível a problemas decorrentes de modificações internas ou externas (feitas em outras aplicações). Dessa forma, a arquitetura da solução deve permitir testes unitários e integrados.

Essa lista de subcaracterísticas tem por objetivo especificar quais são as prioridades de um portal corporativo. Durante a fase de projeto, é necessário analisar uma série de alternativas e ter em mente quais atributos são mais necessários ajuda a basear e justificar essas decisões. Uma característica não tão necessária para um portal corporativo é a portabilidade. Por ser uma ferramenta Web, o portal é instalado apenas no servidor de aplicação, um ambiente bem mais estável que as estações de trabalho que o acessam. Não há grandes preocupações, com usuários usando outros sistemas operacionais, por exemplo, havendo um navegador com acesso à rede, o acesso ao portal se torna possível.

Outro atributo importante para qualquer software, porém menos relevante, se comparado a outros no contexto de um portal, é a eficiência com relação ao tempo. Evidentemente, tudo depende dos requisitos não funcionais especificados, mas os tempos de respostas exigidos são maiores do que os exigidos, por exemplo, para aplicações de tempo real.

3.4. Visão tecnológica

Para dar suporte às funcionalidades de negócio citadas na sessão anterior, soluções de portal corporativo possuem uma série de recursos tecnológicos importantes. Esta sessão do trabalho vai tratar desses recursos e como eles são importantes para disponibilizar as funcionalidades necessárias.

3.4.1. Autenticação e *Single Sign-On* (SSO)

Como destacam Polgar, Bram e Polgar (2006), o processo de autenticação pode ser definido como a verificação da identidade de um usuário através de algumas informações fornecidas por ele. Essas informações são, na maioria das vezes, um nome de usuário e uma senha.

Para aumentar o nível de segurança nesse processo, é muito comum solicitar dados adicionais, como um *PIN code* (código gerado por um dispositivo de segurança individual e que muda de tempos em tempos, como é feito nos sites dos grandes bancos).

Além dos dados fornecidos pelo usuário, muitas vezes, as credenciais do usuário estão associadas a um certificado digital, que precisa ser validado por um servidor independente que presta esse tipo de serviço. *Certisign* e *Verisign* são exemplos de empresas que fornecem esse tipo de solução.

A imagem na figura 4 exibe um resumo das formas de autenticação disponíveis e comumente utilizadas. Além das já citadas, por usuário e senha, certificados digitais e chaves baseadas em hardware, como o *PIN code*, há ainda autenticação baseada em biometria, onde se enquadram o reconhecimento de voz, de impressões digitais e a leitura de retina. Quanto maior for o cruzamento dessas formas de autenticação, mais difícil será para alguém se passar por outra pessoa.

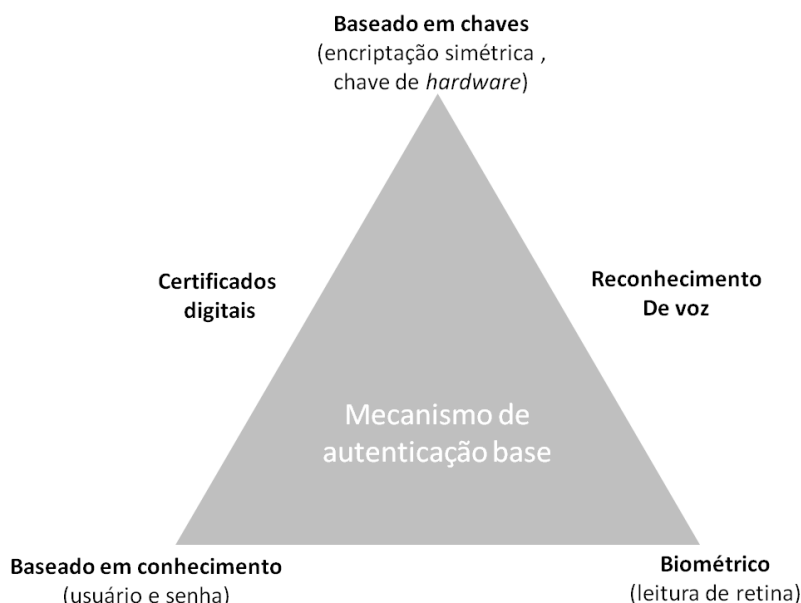


Figura 4. Pirâmide de formas de autenticação, extraída de Polgar, Bram e Polgar (2006).

Como escreveram Tiwari e Joshi (2009), no caso de portais, ocorre um problema nesse processo, já que essas soluções são muitas vezes compostas de várias

aplicações separadas, onde cada uma tem autonomia para exigir o seu conjunto de credenciais. Desse modo, um mesmo usuário, ao acessar o portal, precisa informar credenciais diferentes de acordo com a aplicação que deseja acessar. Ou seja, a cada aplicação que ele tenta acessar, uma nova tela de *login* é apresentada e ele precisa informar as credenciais de acesso para aquela aplicação específica.

Essa situação é bastante desconfortável para o usuário que precisa se lembrar de diversas senhas, uma para cada aplicação a que deseja ter acesso. Muitas vezes, ele opta por anotar as senhas, correndo o risco de que elas sejam descobertas. Além disso, a navegação fica comprometida, pois é necessário passar por diversos processos de autenticação enquanto se usa o portal.

Segundo Tiwari e Joshi (2009), a solução para esse problema é implementar um recurso chamado *Single Sign-On*, onde o usuário passa por apenas um processo de autenticação que garante acesso às aplicações do portal, aumentando a segurança e a navegabilidade do usuário.

Existem diversas formas de implementar esse recurso em um portal com várias aplicações. Nesse trabalho, serão apresentadas três delas: *cookie* compartilhado, *token* temporário e *one-time-password*.

Na opção do *cookie* compartilhado, apresentada por Polgar, Bram e Polgar (2006), a aplicação principal – onde o usuário fornece as credenciais – cria um *cookie* com as informações de identidade do usuário logado. A partir desse momento, sempre que o usuário necessitar acessar uma outra aplicação, ela poderá consultar qual a identidade do usuário logado, acessando o mesmo *cookie*.

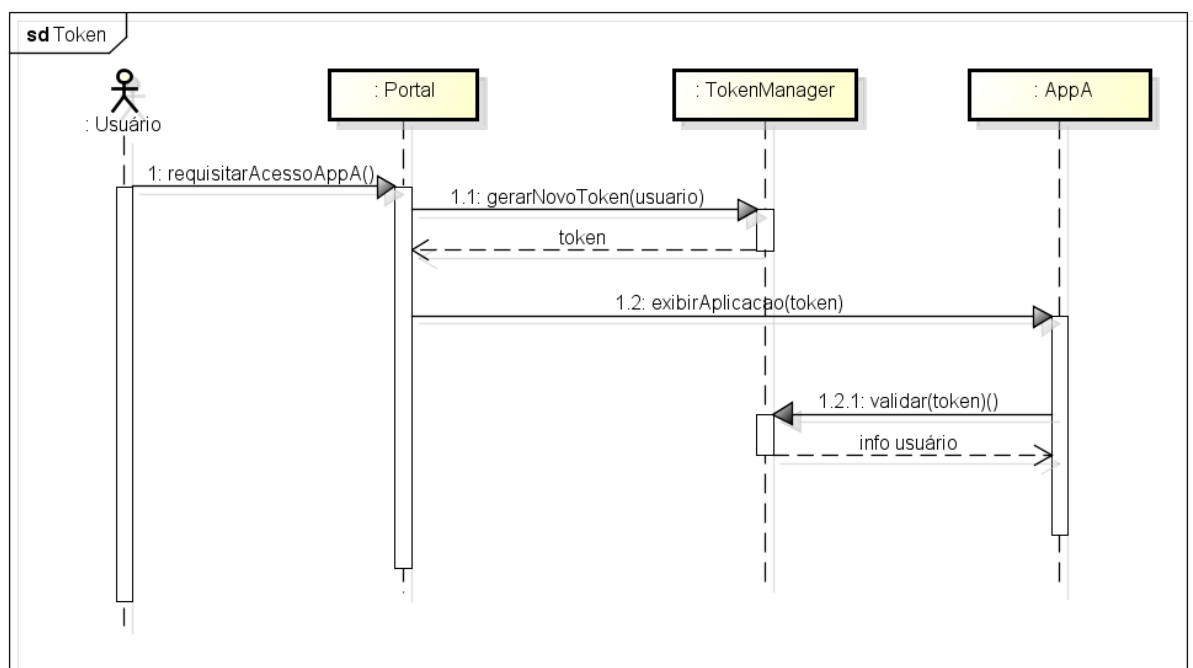
Essa solução é bastante simples de se implementar porém algumas restrições tecnológicas podem inviabilizá-la. É necessário que todas as aplicações que necessitem ler o *cookie* compartilhem um mesmo protocolo para interpretar os dados nele escritos. Essas informações representam a identidade do usuário e são bastante sensíveis, por isso devem ser armazenadas de forma criptografada.

Além disso, de acordo com a RFC 6265, que define os mecanismos de gerenciamento de estado HTTP, *cookies* só podem ser lidos por aplicações

publicadas no mesmo domínio ou em subdomínio da aplicação que o criou. Não é possível, por exemplo, que uma aplicação crie um *cookie* num domínio `http://exemplo.com` e que outra num domínio `http://exemplo2.com` leia o mesmo *cookie*.

Segundo Polgar, Bram e Polgar (2006), uma solução de mercado que implementa essa alternativa é a *WebSphere* da IBM que utiliza *LTPA cookies (Lightweight Third Party Authentication)* para compartilhar as informações de identidade do usuário entre as diversas aplicações que compõem o portal.

Na opção de *token* (ou tíquete), implementada pela *Microsoft* na ferramenta *Microsoft Single Sign-On*, a aplicação principal – onde o usuário informa suas credenciais – gera um tíquete quando o usuário tenta acessar uma outra aplicação e o envia para ela. A aplicação requisitada acessa, através do tíquete, os dados do usuário logado na aplicação. Nesse tipo de solução, um componente é necessário para emitir e validar os tíquetes (MICROSOFT, 2012b). A Figura 5 mostra a troca de mensagens que ocorre nesse processo.



powered by astah

Figura 5. Diagrama de seqüência com fluxo de SSO por tíquete.

O fluxo especificado no diagrama de seqüência é simples, mas exige algumas considerações:

- A comunicação entre o Portal e a Aplicação A (AppA), no passo 1.2, é feita por uma requisição *HTTP* no modo *GET* (protocolo *HTTP*), onde o tíquete é passado como parâmetro. De modo mais prático, nesse momento, o portal exibe a aplicação A, em um *Pop-Up* ou *IFrame*, fazendo uma requisição para uma página dessa aplicação e passando o *token* como parâmetro;
- As trocas de mensagens entre as aplicações e o *TokenManager* são geralmente feitas via serviço;
- Um *token* ou tíquete é geralmente gerado sob demanda e tem uma validade bem curta (cerca de segundos). Ele dura apenas uma requisição, ou seja, depois que a AppA requisita os dados do tíquete, ele se torna inválido, ou seja, mesmo que a aplicação tente obter novamente os dados a partir daquele tíquete, não será possível.

Essa alternativa não tem as restrições do *Cookie* compartilhado e, desde que as aplicações envolvidas tenham acesso ao *TokenManager*, a solução funciona mesmo que elas estejam em domínios separados. Essa mesma solução é utilizada pelo Google, nas APIs que permitem a integração de aplicações com as ferramentas da companhia (GOOGLE, 2012).

Para encerrar a discussão sobre alternativas de implementação de *Single Sign-On*, uma última alternativa chamada de *One-Time Password* foi apresentada por Tiwari e Joshi (2009). A solução utiliza recursos de criptografia e sincronismo para que as duas aplicações comunicantes consigam trocar dados de sessão de forma segura. O fluxo de troca de informações é o seguinte:

1. O usuário informa as credenciais de acesso
2. O portal valida essas informações e permite o acesso
3. O usuário clica em uma aplicação do portal
4. O portal envia para a aplicação o nome do usuário logado
5. A aplicação calcula uma senha para aquele usuário e a envia para o portal

6. O portal também calcula uma senha e a compara com a senha calculada pela aplicação
7. Se as senhas coincidirem, o portal redireciona o usuário para a aplicação

No fluxo proposto, a senha real do usuário nunca é transmitida entre as aplicações. Após a autenticação inicial, todas as trocas são feitas utilizando o nome do usuário e uma senha temporária chamada de *One-Time Password*.

O nome, *One-Time-Password*, remete ao fato de a senha ser válida para apenas uma utilização. Isso é garantido pelo algoritmo de geração que, para calcular as senhas, utiliza um número base que interfere no processo de geração das senhas e que é decrescido a cada utilização. Além do número base, para calcular a senha temporária, as aplicações também utilizam a senha real do usuário (extraída do banco de dados).

É importante ressaltar que o número base é armazenado por usuário. Quando o portal calcula a senha temporária para comparar com a senha calculada pela aplicação, ele decresce o número base. Quando o número base de um usuário chegar a zero, ele deve trocar a senha. A solução proposta é bastante simples de se implementar, inclusive entre aplicações feitas em tecnologias diferentes e rodando em domínios apartados.

3.4.2. Ferramentas de Gerenciamento de Conteúdo

Como definiu Watt (2002), ferramentas de gestão de conteúdo são sistemas desenvolvidos para que empresas consigam criar e gerenciar grandes quantidades de conteúdo, bem como reformatá-lo para uma série de mídias como Internet móvel e TV.

Gerenciar grandes volumes de conteúdo é uma necessidade em grandes companhias que precisam expor, em seus portais, informações detalhadas sobre produtos e serviços prestados. Para suprir essa necessidade, existem ferramentas de gestão de conteúdo que permitem aos usuários sem grande conhecimento técnico criar e publicar informações no *site* da companhia.

Os conteúdos gerenciados nessas ferramentas podem ser textos estáticos, imagens, vídeos, sons e documentos oriundos de dentro da companhia. O que há em comum

nesses conteúdos é que eles mudam muito rapidamente e por isso, os usuários precisam de uma ferramenta que permita atualizá-los o mais rápido possível.

Como as informações publicadas através de ferramentas de CMS podem ser bastante sensíveis, é comum que soluções como estas permitam que um fluxo de aprovação force a revisão das alterações efetuadas por revisores antes da publicação.

As ferramentas de gestão de conteúdo podem ser divididas nas seguintes categorias:

- **Gestão de documentos:** são ferramentas preparadas para armazenar e compartilhar os diversos documentos disponíveis dentro da organização. Ferramentas como essa aceitam diversos tipos de arquivo, como documentos de texto, planilhas eletrônicas, cronogramas, dentre outras coisas. É possível armazenar contratos, manuais e utilizar os recursos de pesquisa da ferramenta para encontrar o que for necessário.
- **Gestão de conteúdo para comércio eletrônico:** ferramentas preparadas para armazenar catálogos de produtos disponíveis para venda.
- **Gestão de conteúdo para Web:** ferramentas preparadas para armazenar e publicar páginas da Web, utilizando os recursos disponíveis nessa tecnologia, como imagens, vídeos e textos. Esse tipo de solução disponibiliza uma interface que permite aos usuários não técnicos trabalhar com esses recursos gerando e atualizando novos conteúdos.

A arquitetura de soluções de gestão de conteúdo é geralmente baseada num paradigma cliente-servidor. Os usuários finais utilizam o navegador para acessar as funcionalidades disponíveis para criação e publicação de conteúdo. Do lado do servidor, há um repositório para armazenar conteúdo (metadados, produtos, páginas e outros itens) e um repositório para armazenar conteúdo não estruturado, como documentos, páginas HTML e imagens. Ainda no servidor, existe o componente responsável por controlar o acesso ao repositório e os fluxos de criação, aprovação e publicação.

Ferramentas como o *Sharepoint* da *Microsoft* permitem a criação de controle de novos tipos de conteúdo, como formulários e registros. Através de soluções como essa, os usuários podem criar formulários para coleta de dados sem grande conhecimento técnico e publicá-los através da Web (MICROSOFT, 2012a).

Segundo Wan et al. (2005), uma nova categoria em sistemas de gestão de conteúdo é a que trata de conteúdos para treinamentos e educação à distância. Recursos como os citados na pesquisa podem auxiliar as áreas de treinamento das organizações na tarefa de capacitar profissionais. Basicamente, ferramentas como as citadas possuem os seguintes recursos:

- Criação de conteúdo para ensino: provê modelos para criação de material, auxiliando no projeto e criação de novos cursos e treinamentos;
- Publicação: provê meios para publicar os materiais através da Web, em formatos de acesso conectado e sem conexão, respeitando o nível em que está o aluno, objetivo do treinamento.
- Função de gestão de conteúdo: provê basicamente as funcionalidades de gestão já discutidas até agora, com o objetivo de manter os registros dos alunos, calendários de cursos e outras informações;
- Apresentação: provê páginas personalizadas para os usuários em vários formatos com HTML, PDF, além daqueles de dispositivos móveis;
- Funcionalidades de comunicação e colaboração: alunos podem utilizar a ferramenta para se comunicar com instrutores e outros alunos. Os instrutores também podem se comunicar com todos os seus alunos através da ferramenta, enviando materiais e alertas.

Na indústria de seguros, uma função de gestão de conteúdo dentro do portal é bastante interessante, pois há, um dinamismo bastante grande nos produtos comercializados pelas seguradoras e conseguir atualizar essas informações sem a interferência da área de tecnologia é fundamental para manter o portal sempre atualizado.

3.4.3. Pesquisa de Documentos

Segundo Chieu, Nguyen e Zeng (2007), grandes empresas acumulam uma grande quantidade de documentos e necessitam de um mecanismo que permita aos colaboradores encontrar os arquivos que necessitam de uma forma ágil. Esse é um recurso bastante importante para um portal corporativo em qualquer tipo de indústria. Em seguradoras, isso também é válido, pois há centenas de contratos, faturas e apólices que precisam ser encontradas em tempo hábil.

Existem algumas formas de se implementar a pesquisa de documentos digitalizados:

- Pesquisa por palavras-chave: ao inserir um novo documento no sistema, o usuário precisa especificar algumas palavras-chave que descrevam o documento e essas palavras são usadas para localizar o arquivo no futuro;
- Pesquisa com base no conteúdo textual: esse tipo de pesquisa é possível em documentos textuais como documentos eletrônicos e planilhas. Nesse tipo de pesquisa, todo o conteúdo do arquivo é analisado em busca dos termos que o usuário necessita encontrar. Esse tipo de pesquisa tem um escopo bem maior do que a busca por palavras-chave;
- Pesquisa com base em semântica: pesquisa que relaciona as palavras contidas nos textos dos documentos com conceitos maiores e mais abrangentes, permitindo, de certa forma, analisar a intenção de um usuário ao pesquisar determinado tópico e assim sugerir resultados.

Outro desafio da pesquisa de documentos é manter a segurança e confidencialidade dos arquivos. Alguns documentos, como contratos, não podem ser acessados por qualquer usuário. Dessa forma, ferramentas que implementam a pesquisa de documentos, geralmente, permitem que sejam definidos os usuários ou grupos de usuários que podem acessar aquela informação.

3.5. Integração de aplicações

Grande parte das funcionalidades de um portal corporativo precisa consumir dados ou serviços de outras aplicações da organização. Um bom exemplo são os *sites* dos

grandes bancos que permitem que o cliente faça uma série de operações pela internet, como transferência e pagamento de contas.

Na verdade, essas operações são implementadas nos sistemas de *back-office* da instituição e o portal serve apenas como um meio de acesso a essas funcionalidades de forma segura e simplificada. As alternativas para se fazer essa interação entre o portal corporativo e as outras aplicações da organização é o assunto dessa sessão do trabalho.

Segundo definiram Hohpe e Woolf (2003), integrações entre aplicações precisam lidar com alguns problemas fundamentais como:

- Rede não confiável: soluções integradas precisam transportar dados de um computador a outro através de uma rede de computadores. Comparado ao transporte em nível de processo dentro de uma mesma máquina, aplicações assim precisam estar preparadas para lidar com uma gama maior de problemas e situações. As aplicações comunicantes podem estar em continentes separados, fazendo com que os dados trafeguem por linhas telefônicas, fibras óticas, roteadores, conexões via satélite, entre outros nós. Cada passo desses pode gerar problemas e atrasos que precisam ser consideradas no projeto de integração;
- Rede lenta: trafegar dados em distâncias grandes entre duas aplicações é muito mais lento que fazer uma chamada local. Projetar uma aplicação distribuída em vários nós como se fosse uma aplicação única pode causar efeitos desastrosos;
- Diferenças entre as aplicações: transmitir informações entre duas aplicações que foram desenvolvidas em tecnologias diferentes, sendo executadas em ambientes diferentes e operando formatos de dados diferentes requer que uma interface comum seja bem definida entre as duas partes;
- Aplicações mudam o tempo todo: uma solução integrada precisa estar preparada para se manter funcionando enquanto as aplicações conectadas evoluem. Uma modificação em uma das aplicações pode resultar em modificações em todas as outras que compõem a solução. Um bom projeto de

integração tem por objetivo minimizar as dependências entre os sistemas conectados utilizando o conceito de baixo acoplamento.

Dentro da disciplina de análise e projeto de sistemas, de acordo com Pressman (2009), acoplamento é uma medida qualitativa do grau em que duas classes são conectadas entre si. À medida que duas classes (e componentes) tornam-se mais interdependentes, o acoplamento cresce. Um objetivo importante de um bom projeto de software é manter o acoplamento o mais baixo possível.

Ainda de acordo com Pressman (2009), a comunicação e a colaboração são elementos essenciais de um sistema orientado a objetos. Há, no entanto, um malefício dessa importante característica. À medida que a quantidade de colaboração e comunicação cresce, a complexidade aumenta tornando o projeto mais difícil de manter e testar.

Do ponto de vista da integração de aplicações o conceito é bem parecido. Ao projetar a comunicação entre dois sistemas, é preciso diminuir as interconexões entre as duas partes, visando um baixo acoplamento da solução como um todo. Complementando a definição de Pressman, de acordo com Hohpe e Woolf (2003), acoplamento é a medida de quantos pressupostos cada parte de uma comunicação precisa fazer do funcionamento da outra parte. Um bom exemplo é uma chamada local dentro de uma aplicação; para que ela funcione as duas partes precisam assumir que o nome do método e a ordem dos parâmetros estão corretos, que os tipos dos parâmetros são compatíveis, que todos os parâmetros essenciais estão preenchidos e concordar quanto ao tipo de retorno. Nessa interação, se qualquer um dos pressupostos falhar, a chamada não ocorre. Muitos dos esforços durante o projeto de um portal corporativo são no sentido de diminuir o acoplamento com as aplicações com as quais o portal se conecta.

Nas próximas seções, a discussão sobre as alternativas existentes será feita com a apresentação de padrões de arquitetura disponíveis na literatura consultada. Como definiu Pressman (2009): “um padrão de projeto descreve uma estrutura de projeto que resolve um problema particular em um contexto específico e em meio a forças que podem ter impacto na maneira pela qual o padrão é aplicado”.

3.5.1. Camada de integração

Para ajudar a contextualizar os portais corporativos dentro do assunto de integração de aplicações, é importante analisar as alternativas arquiteturas que se tem ao projetar a integração, antes mesmo de analisar qual será o método de troca de informações. Segundo Trowbridge e seus colegas (2004), conectar dois sistemas requer basicamente uma infraestrutura que transmita dados de um sistema ao outro. Porém, frequentemente, as soluções projetadas precisam fazer mais do que transportar informações. A camada de integração surge como um componente de ligação que permite que novos recursos sejam adicionados, sem interferir nas duas aplicações comunicantes. É possível adicionar recursos de segurança, automatização de processos, unificação de pontos de acesso e transformação de mensagens. A Figura 6 mostra tal integração.

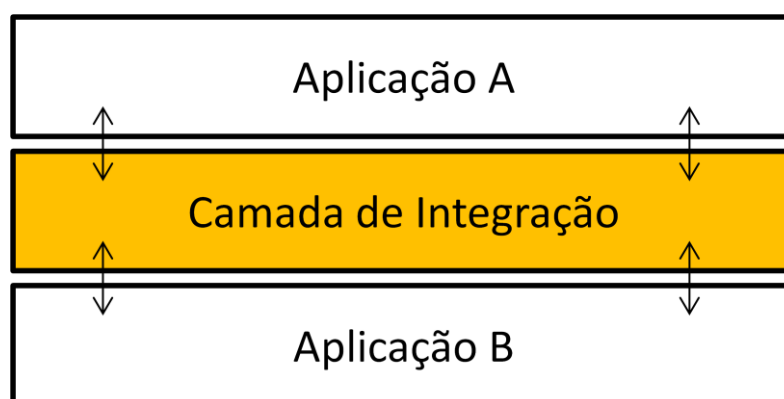


Figura 6. Posicionamento da camada de integração.

Durante o projeto da camada de integração, os seguintes fatores precisam ser considerados:

- **Nível de automação:** no mundo ideal, a troca de informação entre as aplicações de uma organização deveria ser totalmente automatizada, com uma aplicação gerando informações consumidas por outras. Esse nível, porém, é utópico para a maioria das organizações. Uma série de dificuldades já citadas na integração de aplicações faz com que a quantidade de recursos financeiros e de tempo estimados inviabilize tais esforços. Dessa forma, o projeto de integração deve chegar ao nível de automação, onde se consiga aumentar a produtividade dos usuários envolvidos a um custo viável. Muitas vezes, uma tela que exiba dados

de várias aplicações já é suficiente para tal, não sendo necessário que as aplicações interajam diretamente.

- **Nível de abstração e acoplamento:** abstrair os detalhes internos das aplicações durante a comunicação é um ponto chave para que elas possam mudar no futuro sem que haja problemas. Isso impacta diretamente no acoplamento. Como já foi dito, quanto menos pressupostos existirem entre as aplicações, menor o acoplamento. A camada de integração pode ajudar a aumentar o nível de abstração no sentido de que ela mesma pode abstrair alguns detalhes expostos de uma aplicação, não permitindo que eles cheguem até a outra aplicação comunicante. Um exemplo é quando uma aplicação recebe uma série de parâmetros em uma chamada de serviço que não são pertinentes no contexto de uma integração. Nessa situação, a camada de integração pode preencher esses parâmetros sem que a aplicação que precisa consumir o serviço tome conhecimento deles.
- **Manutenção de estado:** muitas soluções de integração são formadas por uma série de passos definidos por um processo de negócio. Nesse contexto, um fator importante é definir quem manterá o estado do processo, já que cada passo é geralmente executado por uma aplicação diferente. A camada de integração é um forte candidato para essa responsabilidade, pois ela pode se comportar como um componente orquestrador responsável por organizar a ordem das chamadas a cada aplicação durante a execução do processo.
- **Dissonância semântica:** uma das grandes dificuldades na comunicação entre duas aplicações são as diferenças entre os modelos de dados das mesmas. Muitas vezes, até campos com o mesmo nome podem significar coisas diferentes, levando a resultados catastróficos quando não identificados. A camada de integração pode ajudar a resolver esses problemas, adicionando recursos de tradução entre os modelos das aplicações.

Considerar esses fatores é primordial para se definir qual estilo arquitetônico a camada de integração deve adotar. Trowbridge e seus colegas (2004) elencaram três padrões nesse sentido: agregação de entidades, integração de processos e

integração via portal. Como o tema deste trabalho visa os portais corporativos, o objetivo das próximas seções é mostrar como as outras duas alternativas podem se juntar à integração via portal. Antes disso, é importante definir cada uma dessas opções.

3.5.1.1. Agregação de entidades

Em ambientes corporativos, repletos de aplicações desenvolvidas em tempos diferentes por equipes diferentes, é muito comum que várias entidades se repitam. Geralmente, a mesma entidade está presente no modelo de várias aplicações, porém com visões diferentes. Ela está, na verdade, particionada nos vários modelos presentes na organização.

Um exemplo dessa situação é a entidade Apólice em uma companhia de seguros. A aplicação responsável por controlar os sinistros da empresa tem as informações de ocorrências de sinistros abertas e indenizadas para aquela apólice. Informações de pagamento como parcelas em aberto ficam a cargo do sistema de cobranças. Informações sobre valores pagos de comissão e agenciamento daquele negócio ficam no sistema de comissão.

Em algum momento, é necessário ter uma visão consolidada de uma entidade particionada. Isso pode acontecer para a geração de um relatório por exemplo. Se nada for feito, a nova aplicação precisará se conectar diretamente aos repositórios das outras aplicações, juntando as partes. Seguindo a classificação de Pressman (2009), esse é um caso de acoplamento de conteúdo, pois uma aplicação acessa dados internos de outra aplicação diretamente, sem nenhum tipo de controle.

Para amenizar o problema, a camada de integração pode fornecer uma versão agregada de entidades-chave dentro da organização, protegendo os repositórios de origem e não permitindo que detalhes dessas aplicações sejam considerados nas aplicações que precisam consumir as informações. No caso da seguradora, ela pode fornecer uma visão completa da entidade Apólice, com as informações provenientes de todos os repositórios. Caso alguma aplicação necessite ver as informações da apólice, pode consumir esse serviço prestado pela camada de aplicação sem ter conhecimento de onde estão os repositórios originais.

A implementação dessa solução pode seguir duas abordagens:

- **Acesso em tempo real:** nesse caso, a entidade agregada não possui um repositório físico de armazenamento, ou seja, ele não possui os dados. Nesse caso, quando uma aplicação requisita alguma informação, ela é buscada no repositório original. Essa abordagem é mais confiável, pois os dados estão sempre atualizados já que o repositório original é acessado. Em contra partida, o método pode ser inviável, dependendo do volume de dados ou de requisições.
- **Replicação:** nesse caso, a entidade agregada possui um repositório físico e os dados são copiados periodicamente dos repositórios originais. Essa abordagem é menos confiável, pois os dados podem não ser os mais atuais, se a periodicidade de replicação for pequena. Em contra partida, possui maior desempenho com grandes volumes de dados.

3.5.1.2. Integração de processos

Um processo de negócio pode ser definido por uma sequência bem definida de passos executados a fim de se atingir um objetivo. No mundo corporativo, muitas vezes, um mesmo processo de negócio conta com a participação de várias aplicações. Um exemplo de um processo de negócio é o fluxo de renovação de uma apólice dentro de uma seguradora. Quando a proposta de renovação chega, o número de ocorrências de sinistros naquela apólice deve ser consultado na aplicação que controla esse tipo de ocorrência com o objetivo de calcular o reajuste, o sistema de cotação precisa calcular o novo valor a ser pago, a situação do cliente deve ser consultada no sistema de cobrança para checar parcelas em atraso, o sistema de comissão deve calcular os valores a serem repassados e assim por diante.

A abordagem mais simples nesse caso é fazer com que cada aplicação, ao terminar sua parte, faça a chamada para a próxima aplicação (próximo passo). Essa abordagem é ruim do ponto de vista da manutenibilidade, pois qualquer alteração na ordem dos passos causa modificações nas aplicações que participam do processo. O problema é que cada aplicação precisa obrigatoriamente conhecer a aplicação

seguinte. Seguindo a classificação Pressman (2009), essa solução está acoplada por controle, pois o controle é passado diretamente de uma aplicação para outra.

Para amenizar o acoplamento nessa solução é necessário adicionar um componente que controle o fluxo de chamadas de acordo com a ordem definida no processo de negócio. Esse componente pertence à camada de integração e é chamado de gerenciador de processo. Basicamente, nenhuma das aplicações envolvidas tem conhecimento uma da outra, apenas o gerenciador as conhece. Quando o processo se inicia, o gerenciador chama a primeira aplicação, aguarda a resposta, depois chama a segunda aplicação e assim por diante.

Uma figura importante nessa solução é o modelo do processo que é uma especificação do processo. Nesse modelo são definidos cada um dos passos, a ordem de execução e como cada um é implementado (qual é o sistema responsável e como será feita a conexão com ele).

O gerenciador de processo recebe como entrada o modelo de processo e a partir dele, consegue gerenciar a execução do mesmo. Além desse recurso primário, há outras preocupações com as quais o gerenciador precisa lidar:

- Execução de múltiplas instâncias do mesmo processo: voltando ao exemplo da renovação de apólice, é possível que várias apólices estejam sendo renovadas ao mesmo tempo, ou seja, há múltiplos processos de renovação acontecendo em paralelo. É papel do gerenciador de processo distinguir corretamente as mensagens de cada instância do mesmo processo.
- Tratamento de erros: num processo longo, com várias aplicações envolvidas, há grande possibilidade de ocorrerem exceções. O gerenciador deve prover meios de coleta dessas ocorrências, bem como formas de tratar os problemas durante o processo com fluxos alternativos.
- Gerenciamento de transação: é responsabilidade do gerenciador de processo manter o ambiente como um todo, num estado consistente, ou seja, quando ocorrem problemas no meio de um processo, deve haver recursos que permitam

que as aplicações que já executaram sua parte desfaçam as modificações efetuadas.

3.5.1.3. Integração via Portal

Existem situações em uma organização em que o usuário precisa acessar informações em vários sistemas para realizar uma operação. Um exemplo é quando um usuário precisa acessar o histórico de compras e pagamentos de um cliente antes de aprovar uma nova compra. Esses dados podem estar em sistemas diferentes e se nada fosse feito, o usuário precisaria acessar os dois sistemas simultaneamente para decidir. Esse processo pode induzir o usuário a erros, caso ele não consulte – devido a um engano - o mesmo cliente nos dois sistemas, por exemplo.

Uma solução é utilizar o conceito de portal, onde um componente na camada de integração serve como agregador do conteúdo das duas aplicações, fornecendo ao usuário informações dos dois sistemas de maneira simplificada. O portal fornece através de uma única interface gráfica informações de várias aplicações, como é possível ver no diagrama a seguir, na Figura 7.

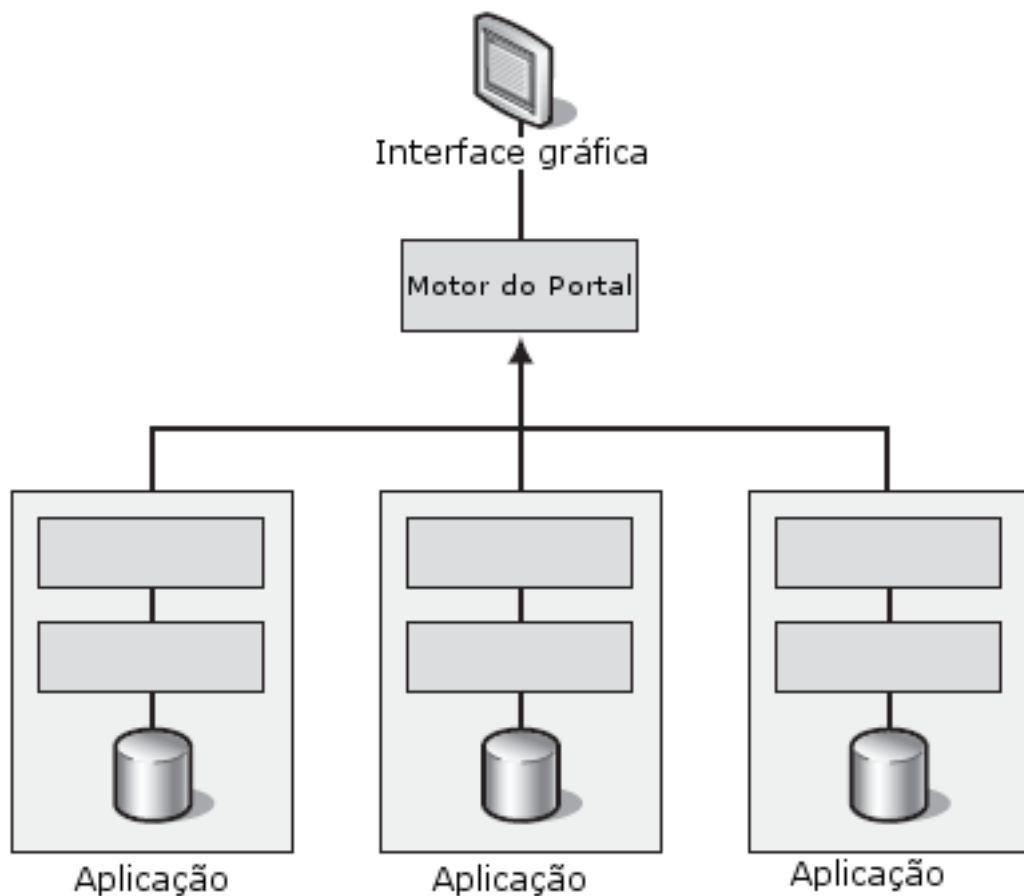


Figura 7. Integração via portal, extraída de Trowbridge et al. (2004).

O motor do portal é um componente especializado em extrair informações de outras aplicações de várias formas e exibi-las nos padrões Web; logo, o acesso pelos usuários é feito via navegador. Quanto ao tratamento que o portal dá às informações extraídas de outras aplicações, existem algumas possibilidades dentro da integração via portal:

- Somente exibição: os dados de várias aplicações são exibidos em diferentes regiões da tela. Cada aplicação tem seus dados exibidos num quadro diferente e não há nenhuma interação com os dados, nem entre os mesmos;
- Pós-processamento simplificado: ao invés de apenas apresentar as informações recebidas de cada aplicação, o portal aplica algumas regras simples para exibir as informações. Por exemplo, se uma lista de parcelas está sendo exibida e uma delas está em atraso, o portal a exibe em vermelho;

- Interação com uma aplicação: portais podem servir para mais do que somente exibir informações de uma aplicação. O usuário também pode interagir com um dos quadros inserindo ou atualizando informações. Como não há interação com os outros quadros da tela, não é possível implementar processos de negócio mais complexos, envolvendo mais de uma aplicação. Uma forma de mitigar essa limitação é compor os padrões apresentados;
- Interatividade entre painéis: algumas formas mais simples de interação entre os vários painéis podem ser aplicadas como exibir dados de uma mesma entidade em todos os painéis. Por exemplo, se o objetivo é exibir as informações de uma apólice selecionada anteriormente pelo usuário, o painel de sinistros pode carregar as informações de ocorrências de sinistro daquela apólice; já o painel de parcelamento pode exibir as parcelas em aberto da mesma apólice e assim por diante. Para que isso seja possível, é necessário que haja uma chave comum entre todos os sistemas, no caso, o número da apólice.

Os modos de interação elencados são bastante simples de implementar, o que é um contraponto com as outras alternativas elencadas para a camada de integração (agregação de entidades e integração de processos). Apesar da simplicidade, uma limitação latente desse tipo de solução é a impossibilidade de se implementar requisitos de negócio mais complexos, com fluxos que envolvam várias aplicações. Voltando ao exemplo já apresentado do novo pedido, se o requisito fosse diferente e o sistema decidisse que o pedido poderia ser aceito consultando diretamente o registro de parcelas, isso não seria possível via portal, já que os quadros não interagem entre si nesse nível.

Para mitigar esse tipo de limitação, é necessário compor a camada de integração com as alternativas elencadas até aqui. Por exemplo, ao invés do portal acessar diretamente a aplicação que contém uma parte dos dados da apólice, ele poderia acessar um repositório de entidades agregadas, onde todas as informações já se encontram disponíveis.

Para implementar um processo de negócio mais complexo através do portal, ele pode acessar um gerenciador de processo provocando a criação de uma nova

instância de um processo pré-configurado. Por exemplo, se a renovação de uma apólice pode ser feita via portal, o motor do portal se comunica com o gerenciador do processo de renovação, que já possui um modelo do processo configurado, que irá instanciar um novo processo (de renovação).

Na Figura 8, o usuário tem acesso a três aplicações utilizando um único ponto de acesso, o portal. Apesar disso, o portal não é sobrecarregado, implementando processos de negócio complexos ou consultas a vários repositórios de dados. Essa tarefa é delegada a outros componentes na camada de integração responsáveis por fazer a comunicação com as outras aplicações:

- Repositório agregado: composto por entidades agregadas, é gerado através de um processo de carga que copia as informações dos repositórios das outras aplicações e os armazena no novo repositório;
- Gerenciador de processos: executa um processo de negócio em que cada passo é executado por uma das aplicações envolvidas;

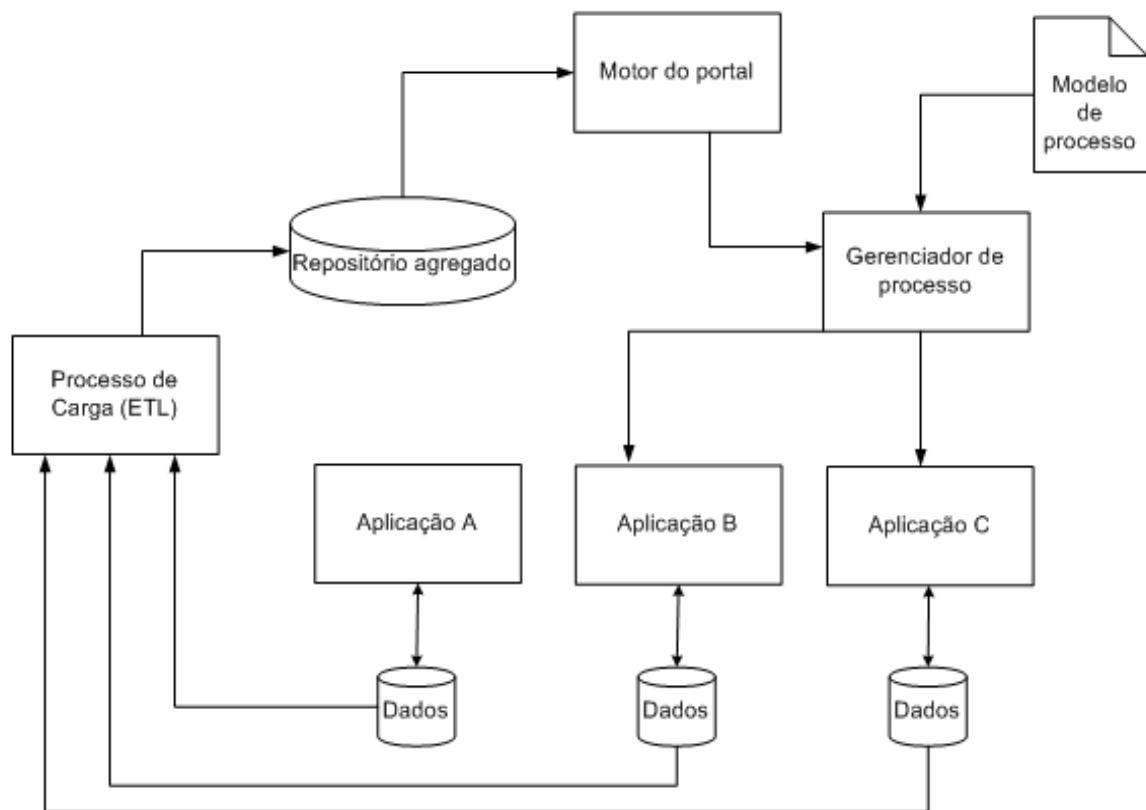


Figura 8. Arquitetura da camada de integração composta pelas três alternativas elencadas.

Essa solução arquitetural mista é interessante também do ponto de vista da reusabilidade. A existência do repositório de entidades agregadas é facilitador não só para o portal, mas para qualquer outra aplicação que precise consultar dados da organização. O ideal seria que esse repositório servisse de base para qualquer aplicação que necessitasse de informações de fora do seu repositório, não sendo necessário acessar repositórios de outras aplicações.

Já a existência do gerenciador, configurado com vários processos de negócio da organização, pode ser utilizado por qualquer outra aplicação que necessite iniciar um desses processos ou mesmo que necessite consultar o status de um processo iniciado, se for uma aplicação de monitoramento.

3.5.2. Métodos de conexão

Como já foi dito, integrar duas aplicações que não foram projetadas para se comunicar é um grande desafio. A camada de integração ajuda a resolver uma parte do problema, mitigando grande parte do acoplamento entre as partes, porém, ainda restam importantes decisões a serem tomadas. É preciso estudar a arquitetura das aplicações envolvidas e definir qual o modo de conexão mais adequado para efetuar a troca de informações. Segundo Trowbridge e seus colegas (2004), a grande maioria das aplicações empresariais opta pelo estilo arquitetural em camadas, que são geralmente três, cada uma delas podendo ainda se subdividir:

- Camada de apresentação: responsável pela interação com o usuário através de interfaces gráficas;
- Camada de regras de negócio: responsável por manter a aplicação num estado consistente, implementando validações e regras que reflitam o contexto em que o software está inserido;
- Camada de acesso a dados: responsável por encapsular o acesso da aplicação aos repositórios de dados usados na aplicação.

Trowbridge e seus colegas (2004) ainda propõem uma classificação para os métodos de conexão disponíveis, baseados na camada onde a conexão ocorre. Se a conexão é feita na camada de regras de negócio, por exemplo, a integração é

funcional. Essa classificação é aplicada para classificar os meios de comunicação apresentados.

Segundo Hohpe e Woolf (2003), existem alguns padrões difundidos para se integrar aplicações, que estão descritos na literatura:

- Troca de informações através de arquivos: é a forma mais simples de todas, quando uma aplicação gera um arquivo que é lido por outra, fazendo a importação dos dados; os autores chamam esse padrão de *File Transfer*;
- Banco de dados compartilhado: nesse método, duas ou mais aplicações consomem o mesmo banco de dados, obtendo acesso às mesmas informações diretamente; Os autores chamam esse padrão de *Shared Database*;
- Chamada remota de procedimento: nesse método, uma aplicação executa uma chamada para outra aplicação; remotamente, a aplicação executa a solicitação e devolve o resultado; os autores chamam esse padrão de *Remote Procedure Invocation*;
- Mensagem: nesse método, uma ferramenta serve de ponte entre as aplicações, permitindo que haja troca de informações entre elas, sem que as duas estejam diretamente conectadas; os autores denominam esse padrão *Messaging*.

As próximas sessões são dedicadas a detalhar cada uma dessas alternativas, analisando pontos fracos e fortes e principalmente levantando os principais pressupostos envolvidos, pois o objetivo maior é reduzir o acoplamento.

3.5.2.1. Troca de arquivos

Nesse padrão, uma aplicação gera um arquivo que é lido por outra aplicação com o objetivo de trocar informações. A troca de arquivos é um dos métodos mais antigos utilizados para troca de informações e tem grande aceitação e abrangência. Um dos motivos é que o conceito de arquivo como uma unidade fechada de *bytes* é reconhecido na grande maioria dos sistemas operacionais de mercado,

possibilitando que aplicações executadas em diferentes ambientes troquem informações sem maiores problemas.

Alguns pressupostos são importantes para que uma integração que usa esse método funcione corretamente. A estrutura do arquivo, o local onde ele deve ser salvo, a periodicidade em que o evento ocorre, entre outras coisas, precisam ser levadas em consideração.

Quanto à estrutura do arquivo, as duas aplicações precisam concordar sobre o formato dos dados que vão utilizar para a comunicação. Para comunicações com arquivos do tipo texto, existem alguns padrões comumente adotados como: arquivos com colunas delimitadas por caractere, arquivos com coluna fixa e arquivos estruturados por metadados, como arquivos XML (*EXtensible Markup Language*).

Com a estrutura do arquivo definida, as aplicações ainda precisam concordar com a semântica dos dados envolvidos. Cada campo a ser enviado requer uma análise no sentido de identificar problemas no entendimento dos dois modelos. Um campo Valor numa entidade Produto pode significar coisas diferentes em duas aplicações e se isso não for identificado previamente, problemas podem ocorrer.

Essa não é uma deficiência da troca de arquivos, mas um problema a ser enfrentado em qualquer método de integração. Como já comentado, a camada de integração pode ser um bom lugar para resolver esse tipo de problema, modificando o arquivo gerado antes de chegar à aplicação de destino.

As aplicações também precisam concordar quanto ao local de gravação dos arquivos. Se as duas aplicações são executadas em um mesmo servidor ou numa mesma rede, o local pode ser um diretório local ou compartilhado. Se as aplicações não estiverem na mesma rede, é necessário utilizar algum outro recurso para transportar o arquivo; um dos mais comuns é usar um diretório FTP (*File Transport Protocol*).

Quanto à periodicidade de geração de arquivos, é necessário que as aplicações envolvidas concordem no momento em que os arquivos estarão disponíveis e de quanto em quanto tempo isso ocorrerá. Algum critério deve ser adotado pela

aplicação que lê os arquivos para identificar arquivos já processados, bem como deve haver uma política para apagar esses arquivos quando eles já tiverem cumprido seu papel.

Ainda sobre a sincronização de tempo na geração de arquivos, recursos de monitoramento do sistema de arquivos incorporados aos *frameworks* de desenvolvimento mais modernos permitem que uma aplicação seja notificada toda a vez que um diretório for alterado. Dessa forma, a aplicação que precisa ler os dados do arquivo pode iniciar seu processamento tão logo a aplicação que escreve termine seu trabalho. Um exemplo de componente que possui esse tipo de serviço é o *FileSystemWatcher* disponível no *Microsoft .NET Framework* (MICROSOFT, 2012c).

Segundo Hohpe e Woolf (2003), a grande vantagem da troca de arquivos é que as aplicações envolvidas precisam conhecer muito pouco sobre os detalhes internos umas das outras, o que leva a solução a um baixo acoplamento. Todos os pressupostos levantados dizem respeito ao arquivo (formato, periodicidade de geração, local de gravação, entre outros), por isso, as aplicações podem mudar internamente sem que isso acarrete problemas, desde que o processo de geração e leitura dos arquivos permaneça funcionando da mesma forma.

Seguindo a classificação de Trowbridge e seus colegas (2004), a troca de arquivos é um tipo de integração em nível de dados, já que as informações estão sendo extraídas do repositório de uma aplicação para um arquivo e depois gravadas no repositório de outra, a partir do mesmo arquivo.

3.5.2.2. Banco de dados compartilhado

Outra alternativa para integração na camada de dados é fazer com que as aplicações que precisam trocar informações usem o mesmo repositório de dados. Como grande parte das aplicações empresariais persiste as informações em bancos de dados, uma alternativa a criar arquivos de extração, é ler as informações da própria base de dados onde elas residem.

A facilidade de conexão das ferramentas de desenvolvimento com os gerenciadores de bancos de dados mais utilizados torna essa alternativa interessante do ponto de

vista da praticidade. Basicamente, a aplicação que necessita dos dados precisará de uma conexão com um banco de dados.

Alguns problemas da troca de arquivos são resolvidos pelo compartilhamento de banco de dados; o principal deles é a latência de atualização. Como o processamento de arquivos ocorre periodicamente, não há garantia de que ambas as aplicações estejam trabalhando com as versões mais atualizadas dos dados. Ao acessar a informação diretamente, o problema é resolvido.

Segundo Hohpe e Woolf (2003), outro problema mitigado é a dissonância semântica existente entre os modelos das aplicações envolvidas. O fato das aplicações serem obrigadas a dividir o mesmo modelo de dados faz com que problemas nesse sentido sejam identificados e resolvidos o quanto antes.

As mudanças necessárias para se chegar a um modelo único e fazer com que as aplicações envolvidas trabalhem com esse modelo tornam esse método de conexão mais intrusivo que a troca de arquivos.

Outra consideração importante é a segurança e a consistência do repositório de dados. Ler dados diretamente da base de dados de outra aplicação não requer grandes preocupações com esses fatores, porém atualizar informações diretamente na base de dados de outra aplicação pode gerar grandes problemas.

Os gerenciadores de bancos de dados permitem algumas inconsistências ao alterar o repositório, como gravar valores nulos em campos obrigatórios, armazenar um valor inválido de acordo com o tipo de campo escolhido, utilizar referências inexistentes às outras tabelas e armazenar campos chaves duplicados. As validações citadas, porém, não são suficientes para manter o repositório no estado consistente esperado pela aplicação proprietária do repositório.

Uma série de validações e consistências é feita na camada de regras de negócio e quando uma aplicação insere dados diretamente na base de dados, todas essas inconsistências deixam de ser avaliadas. Por esse motivo, a adoção desse meio de conexão só é recomendada para leitura de dados.

Outra consideração importante acerca do compartilhamento de banco de dados é a possibilidade de replicação do mesmo. Uma alternativa a acessar a base de dados de uma aplicação diretamente é criar um processo que replique a base dados periodicamente. Esse método tem alguns benefícios:

- **Redução de gargalos:** um problema de se ter várias aplicações acessando a mesma base dados é que a concorrência gera lentidão para todos os envolvidos. Um exemplo clássico é tentar conectar uma aplicação de apoio à decisão (A), que precisa gerar relatórios muito grandes e que cobrem grande parte da base, diretamente à base transacional, onde as aplicações operacionais (B, C etc.) registram toda a atividade da empresa. O resultado é que a lentidão das pesquisas da primeira aplicação (A) atrapalha o processamento de transações das demais aplicações. Além disso, a contenção gerada pelas transações B, C etc., da segunda base pode atrapalhar ainda mais a execução das consultas da aplicação A.
- **Modelagem customizada:** a tarefa de se criar uma nova base de dados gera uma oportunidade para modelar uma base mais adequada às necessidades da aplicação que necessita das informações. Por exemplo, se a aplicação em questão é uma aplicação de apoio à decisão, modificar a granularidade da informação é uma boa prática, pois aplicações desse tipo geralmente trabalham com dados sumarizados.
- **Menos intrusiva:** como uma nova base é criada, não há necessidade de mudança na base de dados original, nem na aplicação que a utiliza. Como disseram Trowbridge e seus colegas (2004), é o modelo ideal para trocar informações em nível de dados com aplicações adquiridas na forma de pacote comercial e que, por isso, não permitem alterações.

3.5.2.3. Chamada remota de procedimento

Muitas vezes não basta compartilhar dados entre aplicações; há casos em que processos implementados dentro de uma aplicação precisam ser utilizados por outra. Por exemplo, o sistema de cotação de uma seguradora implementa todo o processo de cálculo que envolve uma cotação de seguros. Se outra aplicação

precisa realizar o mesmo processo, não é suficiente gravar a nova cotação diretamente no repositório do sistema de cotação (usando o método de base de dados compartilhado). Se isso for feito, todo o processo de cálculo, com suas validações e consistências, já implementadas e testadas, não será reaproveitado.

Esse exemplo ilustra também um dos problemas da integração em nível de dados, já levantado nas seções anteriores, a quebra do encapsulamento dos dados internos da aplicação que possui o repositório, no exemplo, o sistema de cotações. Como disse Hohpe e Woolf (2003), esse é um dos pilares da orientação a objetos, definindo que os dados internos de um módulo (seja ele uma classe ou mesmo um programa) nunca devem ser alterados diretamente por módulos externos. A ideia por trás desse conceito é que qualquer alteração nos dados internos seja feita por um procedimento público na interface do módulo, permitindo assim, que validações sejam feitas antes e depois da atualização, mantendo o estado interno consistente.

A solução para o problema apresentado no exemplo é o sistema de cotação expor o procedimento responsável pelo processo de cotação de modo que outras aplicações possam realizar as cotações, sem acessar diretamente sua base de dados. Desse modo, o encapsulamento é garantido, além de ser possível reutilizar o procedimento já existente.

Existem diversos protocolos e ferramentas no mercado para auxiliar na tarefa de expor procedimentos locais para chamadas de outras aplicações. CORBA (*Common Object Request Broker Architecture*), COM (*Component Object Model*), .NET *Remoting*, Java RMI são exemplos dessas tecnologias. O problema de algumas dessas soluções é que elas não permitem a comunicação de aplicações desenvolvidas em linguagens de programação diferentes. Por exemplo, .NET *Remoting* pode ser usado exclusivamente entre duas aplicações .NET, não sendo possível para uma aplicação Java invocar um procedimento implementado em uma aplicação .NET.

Quando existe a necessidade de interoperabilidade entre aplicações desenvolvidas em linguagens diferentes, uma opção é desenvolver um *WebService*. Esse mecanismo é composto por uma série de métodos expostos através do protocolo SOAP. Segundo o W3C (2012), SOAP é um protocolo baseado em XML que permite

que aplicações troquem informações usando HTTP. O protocolo permite que aplicações de tecnologias diferentes se comuniquem, por exemplo, uma aplicação Java pode invocar um método em um *WebService*, implementado em .NET e vice-versa.

A exposição de serviços tira vantagem do fato de que a maioria das aplicações é desenvolvida em três camadas (Fowler *et al.*, 2002), fazendo com que as regras e validações sejam implementadas pela camada de regras de negócio. Do ponto de vista da camada de interface com o usuário, essa camada disponibiliza uma série de procedimentos responsáveis por aplicar corretamente as consistências que o negócio exige, garantindo que o estado geral do sistema seja mantido correto. Dessa forma, conectar sistemas por chamada remota de procedimento é basicamente expor os procedimentos da camada de negócio para outras aplicações, fazendo com que todas as regras e consistências aplicadas às entradas feitas pela interface com usuário também sejam aplicadas às chamadas remotas.

Um conceito importante por trás desse meio de conexão é a interface do serviço. Uma aplicação expõe através de um dos protocolos e ferramentas citados uma série de procedimentos pelos quais outras aplicações podem se conectar a ela. Os nomes desses procedimentos, os tipos dos seus argumentos e seus tipos de retorno compõem a interface.

Uma vantagem em comparação com a integração por dados é a maior estabilidade da interface do serviço exposta, em comparação com a estrutura interna do repositório de dados durante a evolução da aplicação. Conforme o sistema evolui, a estrutura do repositório de dados é alterada, com novos campos e tabelas sendo criados e campos tendo seus tipos modificados. Esse tipo de mudança pode impactar em outras aplicações que optaram por aplicar a integração via dados, acessando diretamente o repositório.

Por outro lado, a interface de um serviço exposta para outras aplicações tende a permanecer inalterada apesar da implementação do mesmo mudar nas versões posteriores da aplicação. Ainda que haja a necessidade de alteração da interface, ainda é possível usar recursos de programação, como a sobrecarga de métodos,

para manter a interface compatível com versões anteriores, mantendo a integração funcionando sem causar impacto em outras aplicações envolvidas.

A conexão por chamada remota é uma extensão natural do modelo de desenvolvimento. As aplicações são internamente compostas por chamadas locais entre os componentes. Para se integrar com aplicações externas, o modelo é o mesmo, a diferença é que a chamada é remota. De acordo com Hohpe e Woolf (2003), essa pode ser considerada uma desvantagem se o projeto não for cuidadoso. Há grandes diferenças de desempenho entre chamadas remotas e locais e não considerar isso pode trazer grande latência à aplicação.

De acordo com a classificação de Trowbridge e seus colegas (2004), a conexão por chamada remota de procedimento é uma forma de integração funcional, pois as aplicações compartilham funcionalidades e não apenas dados.

3.5.2.4. Mensagem

Um método mais moderno de integração que é capaz de compartilhar dados e funcionalidades é a troca de mensagens. Nesse método, as aplicações comunicantes não se conectam diretamente; essa tarefa fica a cargo de um componente intermediário, geralmente um *middleware* de mercado. Basicamente, uma aplicação envia uma mensagem para o componente intermediário que por sua vez a armazena e, assim que possível, a direciona para a aplicação de destino.

Esse componente, chamado de barramento de mensagens, é o responsável direto por tornar as soluções que o usam menos acopladas do que as soluções que utilizam os outros métodos apresentados até aqui.

Como já foi apresentado, o acoplamento pode ser medido pela quantidade de pressupostos que as duas aplicações precisam fazer antes de se comunicar. Quando a conexão é feita compartilhando a base de dados, temos o pressuposto de que a estrutura interna do repositório não será alterada após a integração, o que nem sempre é verdade. A localização do repositório de dados na rede também é um pressuposto importante, bem como o fato do mesmo estar necessariamente ativo quando a aplicação precisar consumir as informações.

Quando a conexão é feita por chamada remota de procedimento, a interface do serviço desenvolvido para a conexão deve permanecer estável para que a integração continue funcionando. Como no banco de dados compartilhado, a localização da aplicação que fornece o serviço é um pressuposto fundamental. Essa mesma aplicação também precisa estar ativa. Se isso não ocorrer, a integração irá falhar.

Já, se a conexão é feita por troca de arquivos, a periodicidade da troca de arquivos é um dos pressupostos fundamentais, bem como a localização dos mesmos dentro do sistema de arquivos.

O barramento de mensagens consegue eliminar grande parte desses pressupostos:

- Localização das aplicações na rede: como apenas o barramento conhece as aplicações envolvidas, alterações de infraestrutura causam modificações na configuração do barramento e não nas aplicações diretamente.
- Atividade das aplicações e tolerância a falhas: o barramento tem a capacidade de acumular mensagens para uma mesma aplicação quando a mesma não está ativa. Nesse cenário, quando a aplicação envia uma mensagem para uma aplicação que não está ativa, o barramento a armazena e quando a aplicação de destino voltar a funcionar, a mensagem é entregue sem que haja perda de informação.
- Vazão das mensagens: o barramento pode ser configurado para não entregar muitas mensagens de uma vez só para uma mesma aplicação. Nesse cenário, o barramento pode funcionar como um *buffer* que enfileira as mensagens até que a aplicação de destino possa processá-la.
- Formato das mensagens: o barramento também possui recursos de tradução que podem ser usados para mitigar mudanças nos modelos das aplicações. Se uma aplicação A espera uma mensagem de uma aplicação B, num formato X, porém a aplicação B - que a envia - só consegue fazer num formato Y, os recursos de tradução do barramento podem ser usados para adequar a mensagem sem alteração nas aplicações. Isso não é possível quando se utiliza

a chamada remota de procedimento, por exemplo, e exige mudanças na interface do serviço.

Em comparação com a chamada remota de procedimento, uma consideração importante é que o método de programação é alterado na conexão por troca de mensagens. Como já foi dito, o barramento de mensagens se compromete a receber as mensagens, porém a entrega é feita assim que possível. Se a aplicação não estiver ativa, por exemplo, a mensagem pode ser entregue algum tempo depois (quando a aplicação de destino voltar a funcionar). Dessa forma, a aplicação que enviou a mensagem não pode esperar uma resposta imediata, levando à necessidade de programação assíncrona.

Na programação assíncrona, um componente faz uma chamada e não espera a resposta, confiando que a requisição será processada. Opcionalmente, pode haver uma notificação quando a requisição for completada com sucesso ou com falha. Esse é o formato adotado na troca de mensagens e em muitos cenários ele não é possível. Um exemplo de situação como essa é quando há um usuário esperando a resposta de uma tarefa, que depende da execução de uma funcionalidade de outra aplicação. Se a comunicação entre as duas aplicações - a que o usuário está acessando e a que realmente executa a tarefa for assíncrona -, não há como dar uma resposta de imediato para o usuário. Nesses casos, o ideal é que a conexão seja síncrona, para que a resposta da tarefa esteja disponível de imediato, mesmo que a resposta seja, por exemplo, um aviso de que a tarefa não vai ser executada por que um dos sistemas envolvidos está indisponível.

Outra consideração importante sobre a integração por mensagens é a função de roteamento de mensagens, que permite que uma mesma mensagem colocada no barramento por uma aplicação seja retransmitida para várias aplicações conectadas ao mesmo barramento. Esse recurso permite conectar várias aplicações sem acoplá-las diretamente.

De acordo com Hohpe e Woolf (2003), essa capacidade de gerar várias mensagens pequenas torna soluções de integração por mensagens altamente atualizadas. Se um evento em uma aplicação de uma seguradora precisa ocorrer quando uma cotação de seguro for aprovada, a mensagem pode ser gerada e enviada assim que

isso acontecer. Não é necessário aguardar que várias cotações sejam aprovadas para gerar um único arquivo. A velocidade de resposta ainda é menor se comparada à chamada remota de procedimento, mas a aplicação que gera a mensagem não precisa aguardar a execução dos serviços solicitados por outras aplicações similares, mas não correlacionadas, já que tudo ocorre assincronamente.

Seguindo a classificação de Trowbridge e seus colegas (2004), a integração por troca de mensagens se enquadra na integração funcional, pois as informações enviadas visam à execução de funcionalidades no sistema de destino. Se uma aplicação precisa que um processo de outra aplicação seja executado, ela gera uma mensagem no formato de uma requisição, passando todos os parâmetros necessários. A aplicação recebe a requisição do barramento de mensagens, executa a tarefa e envia uma mensagem de resposta.

3.6. Considerações finais do capítulo

Na visão do autor, o capítulo que se encerra tem como ponto principal a discussão sobre integração de sistemas. Esse é com certeza um dos maiores desafios enfrentados na criação de um portal corporativo. De nada adianta termos uma boa estrutura implementada para o portal, se a comunicação com os sistemas envolvidos na em suas funcionalidades não estiver bem projetada.

4. DESENVOLVIMENTO

Nesse capítulo, os conceitos elencados no decorrer do capítulo anterior serão aplicados a um caso real com o objetivo de definir o portal de uma grande seguradora do ramo de vida. O resultado final do capítulo é a arquitetura proposta.

4.1. Projeto de um portal corporativo

O objetivo dessa seção é apresentar a aplicação dos conceitos e padrões descritos no capítulo de Estado da Arte. Para exemplificar, o portal corporativo de uma grande seguradora do ramo de vida é projetado através da especificação de seus módulos. Os requisitos funcionais e não funcionais desse portal são apresentados com o objetivo de apresentar as alternativas ao se atender a cada um deles. Como muitos dos requisitos dependem de aplicações previamente desenvolvidas, soluções para a troca de informações são analisadas. Soluções de segurança, gestão de documentos e conteúdo também são consideradas com o objetivo de montar um portal que atenda às necessidades da organização.

4.1.1. Requisitos funcionais

O quadro 1 apresenta uma lista dos requisitos funcionais a serem implementados no novo portal.

ID	Descrição	Público	Sistemas de <i>back-office</i> envolvidos
RF001	Cotação e acompanhamento de novos seguros individuais e empresariais	B2B	Sistema de cotação PME e Individual
RF002	Aviso e acompanhamento de processos de sinistro	B2B	Sistema de sinistros
RF003	Consulta de ramais telefônicos dos funcionários da empresa	B2E	Aplicação de catálogo de ramais
RF004	Renovação e endosso de apólices existentes	B2B	Sistema de cotação PME e Individual
RF005	Demonstrativo das informações da apólice.	B2C	Sistema de apólices
RF006	Emissão de faturas para pagamento	B2C	Sistema de cobrança
RF007	Movimentação cadastral de itens segurados (pessoas e bens) em apólices existentes	B2B	Sistema de apólices
RF008	Acesso a manuais e procedimentos	B2E	Nenhum, implementado no próprio portal
RF009	Acesso aos relatórios e <i>dashboards</i> sobre utilização dos planos por parte dos clientes em planos empresariais	B2B	Solução de BI
RF010	Serviço de criação de novos usuários para clientes empresarias e individuais	B2E	Sistema de apólices
RF011	Gerenciamento do conteúdo estático do portal por parte de usuários não técnicos	B2E	Implementado pelo próprio portal

Quadro 1. Requisitos funcionais do portal corporativo.

Ao lado de cada requisito, duas informações importantes são elencadas:

- Uma classificação de qual público é atendido por aquela funcionalidade utilizando os conceitos já apresentadas de B2B, B2C e B2E. Esse é um dado importante para agrupar as funcionalidades de acordo com o público-alvo e também para definir como será autorizado o acesso a esses grupos de funcionalidades, já que, por exemplo, empregados estão dentro da rede da empresa, enquanto clientes não estão;

- A definição de quais sistemas de *back-office* estão envolvidos na implementação daquele requisito. Esse é um dado importante para solucionar as integrações necessárias, posteriormente.

4.1.2. Requisitos não funcionais

O quadro 2 apresenta uma lista dos requisitos não funcionais que restringem a implementação do novo portal:

ID	Descrição
RNF001	Usuários externos (clientes e parceiros) serão autenticados através da apresentação de um nome de usuário e senha previamente cadastrados
RNF002	Usuários internos (empregados) serão autenticados através de suas credenciais de rede. Se o usuário estiver autenticado na rede da organização, nenhuma credencial será solicitada. Se o acesso for feito de fora da rede, o usuário precisará informar suas credencias.
RNF003	Restrição de acesso às funcionalidades de acordo com o usuário autenticado no portal.
RNF004	Restrição de acesso às informações disponíveis nas funcionalidades de acordo com o usuário autenticado. Por exemplo, no caso dos <i>dashboards</i> , um usuário do cliente A não pode acessar informações do cliente B.

Quadro 2. Requisitos não funcionais do portal.

4.1.3. Sistemas de *back-office*

O quadro 3 apresenta a lista de sistemas de *back-office* que, de acordo com a lista de requisitos funcionais, terão algum tipo de comunicação com o portal.

Sistema	Descrição
Catálogo de ramais	Pacote comercial fechado adquirido pela organização sem acesso ao código fonte e que tem por objetivo o cadastro e consulta dos ramais dos funcionários.
Sistema de apólices	Sistema Desktop desenvolvido pela própria companhia e que controla o ciclo de vida das apólices seguradas pela companhia.
Sistema de cobrança	Sistema Web desenvolvido pela própria companhia e que controla as cobranças e faturas referentes às apólices da companhia.
Sistema de cotação PME e Individual	Sistema Web desenvolvido pela própria companhia que efetua cálculos de prêmios de novos seguros individuais e empresariais, endossos em apólices existentes e renovação.
Sistema de sinistros	Sistema Desktop desenvolvido pela própria companhia que controla o processo de abertura, regulação e indenização de sinistros.
Solução de BI	Conjunto de relatórios e <i>dashboards</i> criados para exibir indicadores relevantes para a própria companhia, parceiros e clientes. O meio de visualização dos relatórios é através da Web e solução também foi desenvolvida pela própria companhia.

Quadro 3. Sistemas de *back-office* envolvidos.

Na descrição de cada sistema algumas informações importantes são dadas. O ambiente de execução do sistema (Web ou Desktop) é importante para analisar posteriormente qual a melhor forma de acessar os dados provenientes desses sistemas.

Outra informação importante é o responsável pelo desenvolvimento do software. Provavelmente algumas modificações se farão necessárias nessas ferramentas já que elas não foram desenvolvidas com o objetivo de serem publicadas no portal. Uma ressalva deve ser feita nesse ponto: quando se fala em sistema desenvolvido pela própria companhia, estão sendo considerados os sistemas desenvolvidos por consultorias externas, sob a gestão da companhia. O fato é que a organização tem total controle sobre o código fonte dessas aplicações.

Uma consideração importante é que todas essas aplicações estão funcionando hoje e não há uma política de acesso comum em todas as aplicações. Algumas, como o catálogo de ramais são integradas ao cliente de e-mail usado na companhia, não exigindo autenticação adicional por parte do usuário. Outras, como o Cotador (Sistema de cotação PME e Individual) requerem um conjunto de credenciais (usuário e senha) para acesso. Como já foi dito, uma das funções importantes do portal dentro da companhia é fornecer um meio unificado de acesso.

4.1.4. Estrutura básica

As restrições definidas na sessão de requisitos não funcionais servem como base para que sejam tomadas as primeiras decisões do projeto da arquitetura do portal. Antes de pensar em como as funcionalidades serão implementadas é preciso responder algumas questões. Como o usuário será autenticado no acesso ao portal? Como a informação será restringida de acordo com o seu nível de acesso e como as aplicações que serão publicadas através do portal terão acesso a essas restrições? Todas essas respostas ajudam a definir a estrutura fundamental da solução.

As próximas seções servem para definir e especificar os módulos dessa estrutura básica. Nesse trabalho, um módulo pode ser entendido como uma estrutura coesa formada por telas, classes e estruturas de dados responsáveis por atender a um ou mais requisitos.

4.1.4.1. Manutenção de funcionalidades

É o módulo responsável pelo registro de novas funcionalidades para acesso ao portal. No contexto, uma funcionalidade é uma aplicação publicada no portal. Como já foi comentado, um dos grandes objetivos do portal é fornecer uma plataforma única para acesso às várias aplicações da companhia. Uma nova funcionalidade pode ser definida como um nome e uma URL de acesso. Mais detalhes sobre como será feita essa exibição da aplicação serão fornecidos no módulo Acesso às funcionalidades. Esse módulo atende ao requisito RNF003.

4.1.4.2. Manutenção de usuários

É o módulo responsável pelo registro de novos usuários para acesso ao portal. De acordo com os requisitos RNF001 e RNF002, o método de acesso varia de acordo com a origem do usuário. Usuários externos precisam de credenciais (um nome de usuário e uma senha) novas armazenadas no próprio portal, enquanto usuários internos precisam ser associados a uma conta de rede da própria organização, sem a necessidade de novas credenciais.

Para atender aos requisitos RNF003, cada usuário deve estar associado a uma série de funcionalidades. É através do conjunto de funcionalidades preenchido que o portal vai garantir acesso aos recursos, quando o usuário estiver autenticado.

A resolução do requisito RNF004 é um pouco mais complexa. O primeiro passo é analisar as estruturas de dados das aplicações envolvidas e encontrar as entidades que podem servir como item de configuração de acesso aos dados. Por exemplo, a seguradora possui diversos produtos comercializados e a permissão para acessar informações sobre esses produtos, como cotações e apólices, varia de acordo com o funcionário. Existem equipes de venda responsáveis por alguns produtos e que não têm acesso a informações de outros produtos. A partir dessa análise, é possível concluir que a entidade Produto pode ser usada como item de configuração de acesso.

A utilização da entidade resolve o problema de restrição de acesso às funcionalidades B2E, porém as funcionalidades B2C e B2B precisam de um nível de restrição um pouco maior. Um usuário de um cliente da companhia não pode ter acesso a informações de outras apólices que não a sua própria. Com isso, a entidade Apólice também pode ser usada como item de configuração de acesso.

Com essa análise para atendimento do requisito RNF004, conclui-se que cada usuário precisará estar associado a uma série de produtos e/ou apólices com o objetivo de restringir o acesso às informações a que o mesmo tem acesso, ao utilizar as funcionalidades do portal. Uma ressalva importante é que a utilização dessas entidades como itens de configuração é facilitada pela existência de uma chave comum entre as aplicações. Um número de apólice, por exemplo, é o mesmo em

todas as aplicações da companhia. Isso também acontece com o código do produto. Caso não houvesse uma chave única para as entidades, seria necessário um trabalho de padronização ou a criação de tabelas DE-PARA nas aplicações publicadas no portal.

Cada usuário criado pela funcionalidade também precisa ser associado a uma lista de permissões que definem quais tarefas administrativas o mesmo poderá realizar. As permissões disponíveis são:

- Criação e atualização de documentos;
- Criação e atualização de conteúdo;
- Criação e atualização de funcionalidades.

4.1.4.3. Autenticação de usuários

É o módulo responsável por autenticar os usuários que desejam acessar os recursos disponíveis no portal. Como definido no requisito RNF001, para usuários externos, a autenticação será feita mediante um nome de usuário e senha. Já para usuários internos, como definido no requisito RNF002, as credenciais solicitadas são as mesmas que o usuário já utiliza para acessar a rede da organização.

Outro recurso importante do módulo, com o objetivo de atender plenamente o requisito RNF002, é utilizar o recurso de autenticação integrada ao sistema operacional para não exibir as credenciais de usuários internos já autenticados na rede da organização.

4.1.4.4. Acesso às funcionalidades

É o módulo responsável por orquestrar o acesso de usuários autenticados às funcionalidades do portal. Para acessar o módulo, requisitando uma funcionalidade, será usada uma chamada HTTP no modo GET, com a passagem de um parâmetro que indique qual funcionalidade deve ser acessada (exemplo: GET acessarFuncionalidade?id=999, onde 999 é o id de uma funcionalidade).

Para atender ao requisito RN003, o módulo precisa verificar se o usuário autenticado possui acesso ao módulo solicitado antes de exibi-lo. O menu do portal será uma listagens de *hyperlinks* com o endereço do módulo no formato especificado para funcionalidade ao qual o usuário tenha acesso.

Após verificar se o usuário pode acessar a funcionalidade, o processo de exibição da mesma é iniciado. Basicamente, a exibição é feita com o redirecionamento da requisição HTTP para a URL configurada no módulo de manutenção de funcionalidades, com a funcionalidade selecionada.

As funcionalidades são aplicações que não fazem parte do portal e por isso, não têm acesso aos dados do usuário autenticado. Por esse motivo, durante o redirecionamento, o processo de *Single Sign-On* é executado para que a aplicação que será exibida não necessite solicitar novas credencias ao usuário.

A alternativa selecionada para resolver esse problema é o SSO por tíquete, explanada em detalhe na sessão Autenticação e Single Sign-On (sessão 3.4.1). Para atender o requisito RNF004, o TokenManager (citado na explanação) retorna, além do usuário autenticado, os produtos e apólices ao qual o usuário tem acesso. Dessa forma, a aplicação que será aberta pode restringir as informações corretamente.

A justificativa para utilizar essa abordagem é a possibilidade de lidar com aplicações publicadas em domínios diferentes do portal, o que impossibilita a utilização da abordagem de *cookie* compartilhado. A abordagem *one-time password* é interessante, porém não há, no fluxo proposto, uma possibilidade para transmitir à aplicação os itens de configuração de acesso do usuário autenticado.

4.1.4.5. Manutenção de documentos

É o módulo responsável pelo cadastro de novos documentos que serão disponibilizados no portal. Um novo documento deve possuir, além do próprio arquivo, um nome e uma descrição. O módulo atende ao requisito RF011. Alguns documentos têm acesso restrito, por isso, o administrador pode associar ao documento alguns itens de configuração de acesso (produtos e apólices). Dessa

forma, apenas usuários autenticados com acesso a esses produtos/apólices poderão acessar o documento.

4.1.4.6. Biblioteca de documentos

É o módulo responsável pela exibição dos documentos ao qual o usuário tem acesso. O módulo também conta com um recurso para pesquisa de documentos, caso o usuário tenha acesso a muitos documentos. O módulo atende ao requisito RN008.

4.1.4.7. Criação de conteúdo

É o módulo responsável por permitir que o usuário modifique o conteúdo do portal, adicionando novas páginas ou atualizando as que já existem. A edição é feita através de um editor HTML embutido no módulo, permitindo ao usuário adicionar figuras e textos às páginas. O editor permite que usuários não técnicos consigam atualizar o conteúdo. Esse módulo atende ao requisito RF011.

4.1.4.8. Exibição de conteúdo

É o módulo responsável por exibir as páginas criadas pelo módulo de criação de conteúdo. A operação consiste em consultar o HTML cadastrado no módulo de criação para uma página específica e exibi-la. Esse módulo atende ao requisito RF011.

4.1.5. Funcionalidades

Com a estrutura básica do portal definida, chega o momento de analisar os requisitos funcionais e definir quais módulos serão necessários para atender a eles. Como grande parte dos requisitos envolve outras aplicações, alguma análise das integrações será necessária, bem como modificações nessas aplicações. Pensando no custo da implementação, as modificações em outras aplicações causadas pelo portal precisam ser levadas em conta.

4.1.5.1. Cotação de seguros

Para atender aos requisitos RF001 e RF004, o portal precisa interagir com o Sistema de Cotação PME e Individual. Essa necessidade é um pouco mais complexa que apenas exibir uma listagem de informações através do portal. Além da complexidade do próprio processo de cálculo, existe uma série de outros recursos, como consultas e relatórios, que já estão implementados hoje no sistema de cotação e que podem ser disponibilizados no portal.

A complexidade impossibilita a utilização da integração por dados. É inviável, por exemplo, criar uma página no portal que registre novas cotações no repositório do Cotador. Devido à quantidade de regras de negócio envolvidas nesse processo, seria necessário reimplementar muito código ou correr o risco de gerar registros inconsistentes.

Existe a opção da integração funcional, onde o Cotador faria a exposição de seu modelo de objetos através de serviços e novas páginas seriam desenvolvidas no portal para consumir esse modelo. Ainda assim haveria a necessidade de replicar grande parte do código feito nas páginas do Cotador para as novas páginas.

Como já foi dito, o sistema de cotação é um sistema Web, o que leva a uma alternativa menos custosa: publicar as páginas do Cotador como funcionalidades no portal. Ao invés de criar novas páginas semelhantes às do Cotador, é mais simples e rápido adequar o Cotador para o modelo de autenticação *Single Sign-On* adotado e publicar as páginas do próprio Cotador como funcionalidades do portal.

Cabe uma consideração nesse ponto; dependendo de como foram feitas as páginas da aplicação que se deseja publicar, ajustes de layout precisam ser feitos. Cores de fonte, tipos de botões, palheta de ícones, tamanho das páginas podem sofrer modificações para se adequar ao padrão visual adotado pelo portal.

4.1.5.2. Acompanhamento de sinistro

Para atender ao requisito RF002, o portal precisa interagir com o sistema de sinistros da companhia. O processo de aviso de um sinistro é executado quando o

evento coberto por uma apólice - por exemplo, a morte do segurado - ocorre e a corretora de seguros, ou o próprio beneficiário, reclamam a indenização decorrente de uma cobertura. Esse processo é feito com o preenchimento de um formulário para detalhar a ocorrência. Depois disso, a área de sinistros precisa analisar os dados para decidir se a indenização deverá ou não ser paga, o que envolve vistorias, dependendo do caso, e ações judiciais, se houver suspeita de fraude.

Como no caso da cotação de seguros, as funcionalidades implementadas pelo sistema de sinistros são compostas de uma série de restrições e regras de negócio. Muitas dessas funcionalidades não são contempladas pelo portal. Por exemplo, processos judiciais, registro de vistorias e outras operações continuarão sendo feitas pelo sistema existente. No portal, será possível apenas que o corretor ou beneficiário da apólice avise um sinistro e acompanhe o seu processo. Isso leva a um cenário mais confortável, onde será feita apenas uma escrita, durante a abertura do sinistro.

O fato de o sistema de sinistros não ser um sistema Web impossibilita a simples publicação de suas páginas no portal. Será necessário criar páginas no portal que implementem os requisitos, se comunicando com o sistema de sinistros.

A alternativa adotada para essa funcionalidade é a integração funcional. Para a abertura de sinistro será usada uma integração por mensagem com a adoção de um barramento para controlar a entrega de mensagens emitidas pelo portal e enviadas para o sistema de sinistros. A ideia de usar o barramento é manter o sistema de sinistros estável independente do volume de acesso ao portal. Por mais que existam várias requisições para abertura de sinistro, o sistema não precisa responder imediatamente, já que o barramento enfileira as mensagens até ele possa interpretá-las.

Para o acompanhamento dos sinistros, um serviço deverá ser implementado no sistema de sinistro para expor o status de um sinistro. Como o processo é bastante simples, não havendo a necessidade de escrita no repositório, não há a necessidade de usar o barramento por mensagens. O portal precisa apenas se preocupar com a possibilidade de o sistema de sinistros estar fora do ar e nessa situação, exibir uma mensagem amigável para o usuário.

4.1.5.3. Informativo de apólice

Para atender ao requisito RF005, uma funcionalidade que exiba os dados de uma apólice precisa ser implementada. O objetivo é expor uma página no portal que exiba um resumo das informações de uma apólice. As seguintes informações precisam ser exibidas: dados básicos do cliente, corretor, vigência, coberturas, capitais, parcelamento do prêmio pago e histórico de sinistros.

Essas informações estão espalhadas em diversos sistemas da companhia. Esse é o típico caso em que um repositório agregado é bastante útil. Ao invés de acessar vários repositórios para encontrar cada informação necessária, a organização disponibiliza uma base de dados sumarizada em que a entidade Apólice possui todas as informações necessárias. No caso, existe um banco de dados com essas características na companhia que é atualizado todas as noites com os dados de todas as apólices.

Cabe uma ressalva, essa solução é possível porque não é requisito exibir as informações em tempo real. Não há problema, por exemplo, em só exibir um novo sinistro um dia após a sua abertura. Se isso fosse um requisito, outras formas de integração seriam necessárias.

4.1.5.4. Emissão de segunda via de fatura de pagamentos

O atendimento do requisito RF006 é mais um dos casos em que a integração funcional é necessária. Não basta acessar a base de dados do sistema de cobranças, por exemplo, é necessário acessar o método desse programa que gere os boletos das apólices. Esse serviço precisa ser em tempo real, logo, não é possível utilizar o barramento de mensagens, já que quando o usuário solicita a emissão da segunda via, apenas duas respostas são possíveis: o boleto emitido ou uma falha de indisponibilidade. Por isso, a única solução viável é a criação de um serviço de geração de boletos no sistema de cobrança e que será utilizado por uma funcionalidade no portal.

4.1.5.5. Movimentação cadastral

O processo de movimentação cadastral é o processo em que as corretoras informam para a seguradora a movimentação dos segurados nas apólices empresariais. Demissões, admissões e mudanças de cargo são informadas para que o conjunto de registros dos segurados seja atualizado. O sistema de apólices recebe os arquivos fazendo as validações necessárias. Hoje não há um meio único para o envio dos arquivos; pode ser feito por e-mail ou por FTP.

Com a adoção do portal, é possível criar uma funcionalidade para que os corretores providenciem o envio do arquivo ao sistema de apólices e acompanhem o status do processamento do mesmo. Esse é o objetivo do requisito RF007. Como no caso da abertura do sinistro, o barramento de mensagens será usado para encaminhar o arquivo para o sistema de apólices. Um serviço será implementado no mesmo sistema para consultar o histórico dos arquivos enviados e o status do processamento atual.

4.1.5.6. Biblioteca de manuais e procedimentos

A gestão de conhecimento mencionada no requisito RF008 passa por facilitar o acesso à documentação dos processos da companhia. A biblioteca de documentos, já explanada, pode ser usada para atingir esse objetivo. Os usuários responsáveis pela área de organização e métodos da organização podem acessar a biblioteca e armazenar a documentação dos processos para que os outros usuários possam acessar de qualquer lugar.

4.1.5.7. Consulta de ramais

A implementação do requisito RF003 é um pouco mais complexa, pois a aplicação em questão é uma aplicação desktop que não pode ser publicada no portal. Além disso, ela é um pacote comercial e a organização não tem controle sobre o seu código fonte. Não é possível, por exemplo, solicitar a criação de um serviço que retorne os ramais cadastrados. Dessa forma, a alternativa a se adotar é utilizar o padrão de base de dados compartilhado e acessar o repositório de dados da

aplicação diretamente. Uma página será criada e publicada no portal para consultar os ramais.

4.1.5.8. Relatórios e *dashboards* gerenciais

A implementação do requisito RF009 pode seguir a alternativa da publicação da aplicação. Os relatórios são acessados hoje através da Web, por meio de uma plataforma de exibição de relatórios. Será necessária uma alteração nessa ferramenta para implementação do processo de *Single Sign-On* do portal. Feito isso, os relatórios podem ser publicados no portal como funcionalidades.

4.1.5.9. Criação de usuário para clientes

O requisito RF010 é importante, pois novos clientes da companhia (em seguros individuais ou empresariais) precisam de um código de usuário para acessar as funcionalidades do portal. Esse usuário precisa estar configurado corretamente (com a sua apólice associada) para que o mesmo só consiga acessar as informações às quais está autorizado (RNF004). Uma forma de isso acontecer é utilizar o barramento de mensagens para que o sistema de apólices envie uma mensagem ao portal sempre que uma apólice for criada. Essa abordagem é um tanto invasiva, pois exige uma modificação no sistema de apólices; além disso, muitos dos segurados não têm interesse em acessar o portal, o que deixaria a aplicação com vários códigos de usuário criados sem necessidade.

Uma segunda abordagem é implementar uma funcionalidade pública no portal onde o próprio segurado, informando alguns dados, consiga criar seu código de usuário. Com isso, o código de usuário só seria criado quando o segurado demonstrasse interesse pelas funcionalidades disponíveis. Por segurança, o usuário precisa informar CPF/CPNJ e número da apólice ao portal, que verifica então, no sistema de apólices, se existe alguma apólice com aquele número para o CPF/CPNJ informado. Se existir, permite que segurado crie o seu código e já o associa à apólice. O acesso ao sistema de apólices será feito via serviço.

4.1.6. Modelo de arquitetura

A seguir, um diagrama com o esboço das classes necessárias para implementar as funcionalidades é apresentado. A abordagem arquitetural utilizada foi o DDD (*Domain Driven Development*), sendo assim, padrões como Serviço, Entidade e Repositório estão presentes no diagrama. Para uma explanação completa sobre a abordagem, consultar Evans (2011). As classes responsáveis diretamente pela interação com o usuário foram omitidas para facilitar o entendimento.

As classes de serviço servem basicamente como conexão da interface com o usuário com o domínio da aplicação. Logo, os métodos presentes nessas classes representam, em grande parte, as interações do usuário com a aplicação.

Os repositórios servem para encapsular a interação com repositórios de dados, sejam eles pertencentes à aplicação ou repositórios compartilhados (quando a integração por dados é utilizada).

As fachadas servem para encapsular a interação com os sistemas envolvidos com o portal. Chamadas a serviços ou APIs externas são feitas através dessas classes, evitando que tal inteligência se espalhe pelo código.

Uma ressalva importante é que Fachadas e Repositórios são representados apenas como interfaces no modelo proposto. A ideia é que as classes que vão consumir os serviços desses componentes não se acoplem diretamente à implementação dos mesmos. O objetivo é facilitar os testes já que Fachadas e Repositórios são componentes que dependem de sistemas externos (aplicações ou repositórios de dados). Como as classes não estão diretamente acopladas a esses componentes, é possível criar implementações de teste da mesma, que simulem o comportamento esperado dos sistemas externos, facilitando os testes unitários da aplicação.

4.1.7. Resumo das integrações

A seguir apresenta-se no Quadro 4 um resumo das soluções adotadas para as funcionalidades descritas anteriormente. Nesse quadro, não são desconsideradas funcionalidades estruturais do portal, como criação de usuários e biblioteca de documentos, pois elas não dependem de outros sistemas.

Funcionalidade	Solução adotada
Cotação de seguros	Aplicação publicada
Acompanhamento de sinistros	Nova funcionalidade com integração funcional por mensagem e serviço
Informativo de apólice	Nova funcionalidade com integração de dados com repositório agregado
Emissão de segunda de fatura de pagamentos	Nova funcionalidade com integração funcional por serviço
Movimentação cadastral	Nova funcionalidade com integração funcional por mensagem e serviço
Consulta de ramais	Nova funcionalidade com integração de dados com base de dados compartilhada
Relatórios e dashboards gerenciais	Aplicação publicada
Criação de usuários para clients	Nova funcionalidade com integração funcional por serviço

Quadro 4. Funcionalidades no portal e soluções de implementação

A seguir, o quadro 5 define os serviços que precisam ser implementados nos sistemas envolvidos com as funcionalidades do portal:

Sistema	Serviços necessários
Sistema de sinistros	SinistroTO ConsultarStatusSinistro(IdSinistro)
Sistema de cobrança	ParcelaTO[] ConsultarParcelasAtrasadas(IdApolice) BoletoTO EmitirSegundaVia(IdApolice, Parcela)
Sistema de apólices	MovimentacaoCadastralTO ConsultarStatusMovimentacao(IdApolice) SeguradoTO ConsultarSegurado(IdApolice, CPFCNPJ)

Quadro 5. Serviços a serem implementados nos sistemas envolvidos com o portal

4.1.8. Classificação dos componentes

A seguir o resultado de uma classificação dos componentes de projeto do portal. Os componentes são classificados de acordo com a área de negócio. Alguns componentes do portal seriam reaproveitados independente da área de negócio do portal, outros só fazem sentido se o portal for do mercado de seguros.

Componente	Ramo de atividade
ConfiguracaoAcessoTO	Qualquer
Documento	Qualquer
Funcionalidade	Qualquer
IFachadaBarramentoMensagens	Qualquer
IFachadaSistemaApolices	Seguros
IFachadaSistemaCobranca	Seguros
IFachadaSistemaSinistros	Seguros
IRepositorioApolices	Seguros
IRepositorioDocumento	Qualquer
IRepositorioFuncionalidade	Qualquer
IRepositorioRamais	Qualquer
IRepositorioUsuario	Qualquer
Pagina	Qualquer
Permissao	Qualquer
ServicoAvisoSinistros	Seguros
ServicoBibliotecaDocumentos	Qualquer
ServicoEmissaoSegundaVia	Seguros
ServicoFuncionalidade	Qualquer
ServicoGestaoConteudo	Qualquer
ServicoInfoApolice	Seguros
ServicoManutencaoCadastral	Seguros
ServicoRamais	Qualquer
ServicoUsuario	Qualquer
Usuario	Qualquer

Quadro 6. Classificação dos componentes de acordo com a área de negócio

4.2. Visão organizacional

Um ponto importante e muitas vezes deixado de lado durante é o projeto do portal é organização dos papéis envolvidos na posterior manutenção do mesmo. A solução

entregue é composta de várias funcionalidades e informações que não são imutáveis e tendem a sofrer modificações constantes. Uma nova funcionalidade precisa ser adicionada, novos documentos precisam ser disponibilizados, novas páginas precisam ser criadas, entre outras situações.

A tarefa de criação e atualização de documentos disponibilizados na biblioteca do portal fica por conta dos próprios usuários. O módulo de documentos permite que isso seja feito sem a ajuda do pessoal da área técnica. Com o objetivo de tornar o processo mais organizado, o ideal é que um usuário de cada área da companhia receba a responsabilidade de criação e atualização da biblioteca de documentos, centralizando-se nele a tarefa de avaliar os documentos pertinentes e cadastrá-los no portal.

A tarefa de criação e atualização de conteúdo segue um critério um pouco diferente e deve ser centralizado no pessoal de marketing da companhia para que seja mantida uma consistência no layout das páginas. O módulo de criação de conteúdo contempla a operação por usuários não técnicos, sem conhecer HTML por exemplo. Porém, se as modificações forem muito complexas, como por exemplo, modificações na estrutura das páginas, adição de conteúdos multimídia menos comuns (que não sejam vídeos e imagens), haverá a necessidade de auxílio de alguém da área técnica.

Com relação à criação de novas funcionalidades, a tarefa sempre passará pela área técnica. Os estudos de levantamento das necessidades, entendimento dos sistemas externos envolvidos e projeto técnico feito para as funcionalidades entregues precisa se repetir quando uma nova funcionalidade for solicitada.

A tarefa de criação de usuários também é responsabilidade da área técnica que receberá as solicitações de criação de novos usuários e executará as tarefas usando o módulo de Criação de usuários do portal. Uma exceção é feita aos usuários dos clientes que podem ser cadastrados pelo módulo de Criação de usuários para clientes, onde os próprios clientes podem criar seus *usernames* e senhas.

4.3. Considerações finais do capítulo

Na visão do autor, o capítulo que se encerra tem como ponto principal a apresentação das decisões e suas justificativas durante o projeto do portal. A idéia é facilitar o entendimento do processo proposto através da análise de cada decisão importante durante o projeto.

Além disso, durante a elaboração dos trabalhos constatou-se que muitas das decisões tomadas com relação ao portal de uma empresa de seguros se prestam à definição de arquiteturas de portais destinados a outros tipos de atividades. Dessa forma, após a geração do Modelo de Projeto apresentado na Figura 9, procedeu-se à classificação dos componentes daquele diagrama entre específicos de um portal do ramo de negócio de seguros e os de qualquer outro ramo de negócio. Obteve-se assim o Quadro 6, onde se destacam as especificidades do ramo de seguros. De posse desse Quadro e do Modelo de Projeto pode-se colaborar no projeto de portais para outros ramos de atividades, reutilizando-se uma parte considerável dos trabalhos e decisões aqui apresentadas.

Procurou-se também elencar as necessidades de natureza organizacional, determinando-se um conjunto de papéis e responsabilidades que deverão ser considerados durante e após a implantação do portal. A parte maior dessas responsabilidades é atribuída à área técnica, porém existem tarefas que são atribuídas aos usuários ou ao setor de marketing também.

5. ANÁLISE DE RESULTADOS

O presente capítulo visa analisar o impacto da implementação do portal projetado na companhia de seguros de vida escolhida para a análise.

5.1. Benefícios esperados

Abaixo uma lista de benefícios que a implementação pode alcançar:

- O portal representa um ponto único onde os colaboradores da empresa podem buscar documentos e informações corporativas. Com essa nova fonte de informações centralizada, muitas das trocas de e-mail deixam de ser necessárias, assim como cópias de arquivos importantes espalhados pela rede da companhia sem versionamento e backup adequado;
- O portal gera uma plataforma para publicação de novas aplicações desenvolvidas. Com o novo portal funcionando, qualquer nova aplicação Web dentro da companhia pode ser projetada para funcionar dentro do portal, utilizando seus recursos de segurança e controle de acesso.
- O portal representa um espaço de trabalho virtual para os usuários, onde através de um único ponto de acesso são disponibilizados aplicações e recursos que antes eram acessíveis por vários pontos diferentes e sem grande integração.
- Os recursos do portal direcionados à parceiros e clientes representam um ganho de qualidade no atendimento prestado a esses públicos. Além disso, a possibilidade do auto-atendimento, em que a companhia não precisa desprender recursos humanos para auxiliar, representam uma boa diminuição dos custos operacionais.
- Um benefício indireto das integrações do projeto de um portal são os ativos reutilizáveis gerados, como serviços e repositórios agregados. Para atender aos requisitos que precisam ser entregues pelo portal, alguns serviços precisam ser implementados nos sistema de *back-office* envolvidos. Esses serviços podem ser reutilizados por outras aplicações no futuro.

5.2. Dificuldades esperadas

O processo de projeto e implantação do portal pode enfrentar algumas dificuldades; abaixo apresenta-se uma lista daquelas que já são esperadas desde o início:

- As funcionalidades do portal dependem, em sua maioria, de aplicações de *back-office* que já estão em uso na companhia. Essas aplicações foram desenvolvidas e são mantidas por empresas de desenvolvimentos contratadas pela companhia. O próprio portal também seria desenvolvido por uma empresa de desenvolvimento. Dessa forma, um grande desafio para a companhia seria gerenciar esse grande projeto num cronograma envolvendo tantas empresas.
- Qualquer modificação nas aplicações envolvidas pode representar um risco proporcional ao tamanho da modificação. As empresas responsáveis pela manutenção de cada aplicação precisam garantir que as aplicações continuarão funcionando após o desenvolvimento das integrações com o portal. Além disso, para o usuário final, o portal é uma única aplicação (ele não tem como saber onde termina o portal e onde começa a aplicação de *back-office*) e problemas nas integrações que resultem em dificuldade de uso podem fazer com que a confiança na nova solução diminua.
- Uma dificuldade comum a projetos Web é a questão da usabilidade. Como já foi dito, o portal em questão será acessado por diversos perfis de usuário. Seguindo os processos de engenharia de usabilidade, esses perfis precisariam ser analisados, com o objetivo de elencar, para cada um, o grau de conhecimento médio, dificuldades comuns, média de idade, entre outros dados. Com esse levantamento seria possível analisar as alternativas de *interface* gráfica disponíveis para cada perfil. Esse trabalho é crucial para que o resultado final seja um portal com recursos fáceis de serem entendidos e utilizados.

6. CONSIDERAÇÕES FINAIS

Este capítulo visa encerrar o trabalho apresentando as conclusões da monografia, as contribuições em termos de pesquisa resultantes, além de apresentar algumas sugestões de trabalhos futuros na mesma linha de pesquisa.

6.1. Conclusão

O trabalho atingiu os objetivos iniciais. Através do processo apresentado, pode-se chegar a uma arquitetura capaz de resolver os problemas elencados durante a pesquisa, se preocupando com os atributos de qualidade mais relevantes. O estado da arte apresentado, com base em pesquisas na literatura disponível, foi de suma importância para basear as decisões tomadas no desenvolvimento do projeto.

6.2. Contribuição

A contribuição principal dessa monografia foi apresentar um método de trabalho para as fases de análise e projeto de um portal, baseados no estado da arte do assunto e no conhecimento prático do autor em projetos desse tipo. Através desse trabalho, o leitor tem acesso às tarefas realizadas em cada passo desse processo, bem como as alternativas analisadas e as justificativas de cada decisão, chegando assim a um histórico de projeto que pode ser de grande valor, não apenas para projetos no mercado de seguros, mas também para outros mercados.

A classificação dos componentes que fazem parte da arquitetura proposta para o portal - quanto à possibilidade de reutilização de cada um dos componentes em outros portais de mercados diferentes do de seguros - também representa uma contribuição importante visto que esses componentes, bem como as soluções adotadas no projeto dos mesmos podem ser reaproveitados com facilidade.

6.3. Trabalhos futuros

Uma ramificação importante que poderia ser criada a partir desse trabalho é a formalização de um método para o projeto e desenvolvimento de portais corporativos. O método poderia, com base no conhecimento disponível sobre metodologias de desenvolvimento somado ao conhecimento disponibilizado neste e em outros trabalhos sobre portais corporativos, elencar os processos necessários

desde a identificação das funcionalidades até a entrega final, passando pela identificação dos públicos-alvo, o projeto das integrações e testes.

Outro ponto que não foi tratado nessa pesquisa devido à complexidade e à falta de ferramentas no presente momento é a Web Semântica. Essa disciplina pode, no futuro, impactar positivamente nos módulos de pesquisa de conteúdo e de documentos, permitindo a implementação de novas formas de analisar os filtros que o usuário provê.

REFERÊNCIAS

Associação Brasileira de Normas Técnicas. **NBR ISO/IEC 9126-1. Engenharia de Software - Qualidade do Produto, Parte 1: Modelo de qualidade.** Rio de Janeiro. Junho de 2003.

BARTH, Adam. HTTP State Management Mechanism. Request For Comment 6265. Internet Engineering Task Force. April 2011. Disponível em: <<http://tools.ietf.org/html/rfc6265#section-4.1.2.3>>. p. 10-11.

CHIEU, Trieu C; NGUYEN, Thao; ZENG, Liangzhao. Secure Search of Private Documents in an Enterprise Content Management System. In: IEEE International Conference on e-Business Engineering, 2007. ICEBE 2007. October 24-26. 2007. **Proceedings.** P. 105-112.

EVANS, Eric. **Domain-Driven Design - Atacando as Complexidades no Coração do Software 2ª edição.** Alta Books Editora. 2011. ISBN 9788576085041.

FOWLER, Martin; RICE, David; FOEMMEL, Matthew; HEATT, Edward; MEE, Robert; STAFFORD, Randy. Layering. In: **Patterns of Enterprise Application Patterns.** Addison Wesley. 2002. ISBN 0-321-12742-0.

Gartner Group. Gartner IT Dictionary. Disponível em: <<http://www.gartner.com/it-glossary>>. Acessado em: 22 de julho de 2012.

Google. AuthSub for Web Applications. Google Developers. Disponível em: <<https://developers.google.com/accounts/docs/AuthSub>>. Acessado em: 22 de julho de 2012.

HOHPE, Gregor; WOOLF, Bobby. Introduction. In: **Enterprise Integration Patterns. Designing, Building and Deploying Message Solutions.** Addison Wesley. 2003. ISBN 978-0321200686. p. 10-20.

Microsoft. Records management planning (SharePoint Server 2010). Enterprise content management planning. Publicado em 12/05/2010. Disponível em: <<http://technet.microsoft.com/en-us/library/ff363731>>. Acessado em: 29 de julho de 2012. (A)

Microsoft. SSO Tickets. Understanding Enterprise Single Sign-On. Disponível em: <[http://msdn.microsoft.com/en-US/library/aa754727\(v=bts.10\)](http://msdn.microsoft.com/en-US/library/aa754727(v=bts.10))>. Acessado em: 15 de julho de 2012. (B)

Microsoft Biblioteca MSDN. Classe FileSystemWatcher. Disponível em: <<http://msdn.microsoft.com/pt-br/library/system.io.filesystemwatcher.aspx>>. Acessado em: 22 de outubro de 2012. (C)

PRESSMAN, R. **Engenharia de Software 6ª Edição.** Markron Books. 2009. ISBN 9788563308009.

POLGAR, J.; BRAM, R. M.; POLGAR A. **Building and Managing Enterprise Wide-Portals**. Idea Group Publishing. ISBN 1-59140-662-5. p. 119-133.

SERHANI, M. A.; AL-QIRIM, N.; BENHAREEF, A. Enterprise Services (Business) Collaboration using Portal and SOA-Based Semantics. IEEE International Conference on Digital Ecosystems and Technologies. 4th. 2010. 12-15 April 2010. Dubai, United Arab Emirates. **Proceedings**. p. 450-455.

TANG, Xiao-xin e CHEN, Guo-hua. Construction of Enterprise Information Portal in Cigarette Enterprise. In: International Conference on Management Science & Engineering, 14th. 2007. August 20-22, 2007. Harbin, P.R.China. **Proceedings**. LAN Hua; YANG Yu-hong (Ed.). Harbin, China: Harbin Institute of Technology Press. 2010. P. 293-296.

TIWARI, Paras Babu; JOSHI, Shashidhar Ram. Single Sign-on with One Time Password. Internet, 2009. In: First Asian Himalayas International Conference on Internet AH-ICI2009. November 3-4, 2009. Kathmandu, Nepal. **Proceedings**. p. 1-4.

TROWBRIDGE, David; ROXBURGH, Ulrich; HOHPE, Gregor; MANOLESCU, Dragos; NADHAN, E.G. **Microsoft Patterns & Practices. Integration Patterns**. Microsoft Corporation. 2004. ISBN 0-7356-1850-X. 401 p.

WAN, Liyong; ZHAO, Chengling; LIU, Qingtang; SUN, Junyi. Work in Progress-An Evaluation Model for Learning Content Management Systems: from a Perspective of Knowledge Management. In: Frontiers in Education, 2005. FIE '05. Proceedings 35th Annual Conference. October 19-22. 2005. The Westin Indianapolis, Indianapolis, Indiana, USA. **Proceedings**. p. F3G.

WATT, Dougal. Content Management Systems. In: E-business implementation: A Guide to Web Services, EAI, BPI, e-commerce, content management, portals and supporting technologies. Butterworth-Heinemann. 2002. ISBN 0 7506 5751 0. p. 82-93.

WU, Ying-liang; WANG, Xue. A Knowledge Management Solution Based On Enterprise Information Portal. In: International Workshop on Intelligent Systems and Applications, 2009. ISA 2009. 23-24 May 2009. Wuhan, China. **Proceedings**. Wuhan, China: Hubei University of Technology. 2009. p. 1-4.

W3C. W3Schools. SOAP. Disponível em: <<http://www.w3schools.com/soap/default.asp>>. Acessado em: 15 de novembro de 2012.

GLOSSÁRIO

As descrições a seguir foram extraídas de fontes diversas, conforme citações.

[.NET Remoting] – tecnologia Microsoft que permite a comunicação entre aplicações .NET que residam num mesmo computador ou mesmo em máquinas diferentes conectadas por uma rede.

[Agenciamento] – valor monetário pago à pessoas físicas jurídicas que intermediam negócios de seguro.

[Apólice] – compromisso da seguradora com o segurado que garante o pagamento de uma indenização caso algum evento coberto ocorrer.

[ASP] – *Microsoft® Active Server Pages* é um ambiente de scripting *server-side* (que roda do lado do servidor) que pode ser usada para criar e rodar aplicações *Web* dinâmica e interativas. *Microsoft, 2012.* <<http://msdn.microsoft.com/en-us/library/aa286483.aspx>>. Acesso em 30/05/2012.

[Back-office] – Sistemas utilizados internamente dentro da organização e que suportam principalmente atividades operacionais.

[Content Management System - CMS] – sistema que permite a criação e alteração de conteúdo em tempo real de conteúdos (textos, imagens e vídeos) em páginas de internet sem a necessidade de programação.

[Corretor] – profissional responsável por auxiliar um segurado no relacionamento com a seguradora. No Brasil, ele é obrigatório em qualquer seguro. Ele auxilia o segurado a escolher uma seguradora, tira dúvidas dele com relação a pagamento e o auxilia em caso de sinistro.

[Cookie] – mecanismo utilizado no protocolo HTTP para armazenar informações na máquina cliente. Recurso bastante utilizado em *sites* para armazenar informações que devem ser recuperadas entre as várias requisições e respostas que acontecem entre um cliente e o servidor durante a utilização de um *site*.

[Component Object Model – COM] – Microsoft COM é uma tecnologia disponíveis para a família de sistemas operacionais Windows que permite que componentes se comuniquem. *Microsoft, 2012* <<http://www.microsoft.com/com/default.mspix>>. Acesso em 01/08/2012.

[Common Object Request Broker Architecture – CORBA] – arquitetura e infraestrutura aberta e independente de fornecedor que aplicações de computadores usam para se comunicar. *Object Model Group, 2012.* <<http://www.omg.org/gettingstarted/corbafaq.htm>>. Acesso em 19/12/2012.

[Customer Relationship Management - CRM] – disciplina que cuida do gerenciamento e fidelização da relação com os clientes de uma companhia. Softwares responsáveis por manter os históricos de transações, preferências de cada cliente auxiliando a organização em novas campanhas são softwares de CRM.

[Comissão] – valor monetário pago à pessoas físicas jurídicas que intermediam negócios de seguro.

[Cotação] – orçamento, sem compromisso de ambas as partes, do valor a ser pago por um segurado à seguradora em uma possível apólice.

[Dashboard] – são mecanismos de relatório que agregam e exibem uma série de métricas e indicadores de desempenho permitindo o monitoramento e análise de processos de forma rápida. As métricas são exibidas num formato visual através de gráficos, velocímetros e semáforos tornando a análise bastante intuitiva. *Gartner Group IT Dictionary*. 2013. . Acesso em 03/01/2013.

[Enterprise Application Integration - EAI] – padrões para a integração de aplicações corporativas. Envolve técnicas como troca de arquivos, serviços, integração por fila e outros pontos.

[Enterprise Information Portal - EIP] – aplicação web que reúne em um único ponto de acesso conteúdos agregados de diversas fontes dentro de uma organização.

[Enterprise Resource Planning - ERP] – é definido como a capacidade de oferecer um conjunto integrado de aplicativos de negócios. Ferramentas de ERP compartilham um processo e um modelo de dados comum, cobrindo operações e processos de ponta-a-ponta, tais como aqueles encontrados em finanças, RH, distribuição, manufatura, serviços e cadeia de abastecimento. *Gartner Group IT Dictionary*, 2012. <<http://www.gartner.com/it-glossary/enterprise-resource-planning-erp/>>. Acesso em 03/07/2012.

[File Transfer Protocol – FTP] – protocolo TCP/IP usado para acessar um servidor remotamente, listar seus diretórios e copiar arquivos. Provê autenticação por usuários, pesquisa em diretórios e transferência de arquivos. *Gartner Group IT Dictionary*, 2012. <<http://www.gartner.com/it-glossary/ftp-file-transfer-protocol/>>. Acesso em 03/07/2012.

[HTTP] – protocolo de utilizado na interação entre navegador e Servidor Web, através da porta 80 para troca de arquivos enquanto um site é acessado.

[Java RMI – Remote Interface Invocation] – abordagem da tecnologia Java para prover as funcionalidades de uma plataforma de objetos distribuídos. Ivan Luiz Marques Ricarte, DCA/FEEC/UNICAMP, 2002. <<http://www.dca.fee.unicamp.br/cursos/PooJava/objdist/javarmi.html>>. Acesso em 03/08/2012.

[Knowledge Management] – um programa formal para gerenciar os ativos intelectuais de uma organização. *Gartner Group IT Dictionary*, 2012. <<http://www.gartner.com/it-glossary/km-knowledge-management/>>. Acesso em 03/06/2012.

[Latência] – medida da responsividade de uma rede que é geralmente expressa como o tempo de ida e volta (em milissegundos), isto é, o tempo entre o início de uma requisição através da rede e o recebimento de uma resposta. *Gartner Group IT Dictionary*, 2012. <<http://www.gartner.com/it-glossary/latency/>>. Acesso em 20/06/2012.

[Middleware] – software que auxilia programas e bancos de dados (que podem estar em computadores diferentes) a trabalharem juntos. *Gartner Group IT Dictionary*, 2012. <<http://www.gartner.com/it-glossary/middleware/>>. Acesso em 27/06/2012.

[Modelo de Dados] – modelo que especifica como os dados de uma aplicação são estruturados. Nesse modelo, são definidas as entidades que compõe o sistema, seus atributos e relacionamentos.

[PHP] – é uma linguagem de *scripting* que roda no lado do servidor (*server-side*) e pode ser usada para construir aplicações Web acessando bancos de dados e arquivos. O PHP é uma linguagem livre e de código-fonte aberto.

[RFC 6265] – é um documento, elaborado pelo Internet Engineering Steering Group (IESG), que define os campos HTTP Cookie e Set-Cookie do cabeçalho de agentes de usuário HTTP, de forma a que os servidores HTTP possam aí manter o estado de uma sessão quando se usa o protocolo HTTP.

[Seguradora do ramo de vida] – na linguagem da área de seguros, engloba a parte relativa a saúde, previdência e demais itens relativos ao indivíduo.

[Single Sign-On] – serviço que fornece ao usuário a possibilidade de se autenticar em várias aplicações em um único processo de *login*.

[Sinistro] – evento que desencadeia o pagamento de uma indenização por parte da seguradora. Ex.: roubo de um carro segurado por um seguro de automóveis, morte de uma pessoa segurada em um seguro de vida, entre outros.

[Supply Chain Management – SCM] – disciplina que cuida do gerenciamento de toda a cadeia de suprimentos de um produto. Para se produzir um carro, por exemplo, são necessários produtores de materiais básicos, como aço, de componentes mais complexos (como a embreagem), da montadora, do distribuidor e por fim, da concessionária. Software que gerenciam as relações entre cada ponto, como pedidos, entregas, pagamentos, são softwares de SCM.

[Tabela DE-PARA] – tabela que possui uma série de registros de mapeamento de códigos de uma entidade. Quando uma entidade possui uma codificação diferente em duas aplicações, uma tabela como essa é necessária para possibilitar o

entendimento por parte das aplicações envolvidas de que dois registros com códigos diferentes representam, na verdade, o mesmo item no mundo real.

[Uniform Resource Locator - URL] – cadeia de caracteres que identifica os exatos nome e localização de um documento na internet. *Gartner Group IT Dictionary*, 2012. <<http://www.gartner.com/it-glossary/url-uniform-resource-locator/>>. Acesso em 21/07/2012.