

**Criação de uma solução conceptual que permita uma  
integração entre ERPs e Redes Sociais**

André Filipe Costa Restolho

**Projeto submetido como requisito para obtenção do grau de Mestre  
em Gestão de Serviços e da Tecnologia**

Orientador:

Prof. Doutor Carlos Manuel Jorge da Costa, Prof. Auxiliar, Escola de Tecnologia e  
Arquitetura, Departamento de Ciências e Tecnologias de Informação

Outubro 2014

## **Agradecimentos**

Agradeço ao meu orientador, Professor Doutor Carlos Manuel Jorge da Costa por toda a sua orientação, apoio e disponibilidade prestados durante a realização desta dissertação.

Agradeço ao Dr. Fernando Bento, Diretor Técnico da RISA Consulting por todo o apoio e disponibilidade e ferramentas fornecidas, pois sem isso o projeto não teria sido possível de realizar.

Aos meus amigos e colegas, Diogo Lopes, Duarte Fidélis, Fábio Ferrão, Marta Madeira e Raquel Espírito Santo e a todos os membros da 'Situation' que estiveram ao meu lado durante esta fase, pela força e apoio em todos os momentos.

Por fim agradeço à minha família o interesse, apoio, motivação e compreensão transmitida no decorrer da realização desta dissertação.

# Índice

<b>Agradecimentos</b> .....	<b>ii</b>
<b>Índice</b> .....	<b>iii</b>
<b>Índice de Figuras</b> .....	<b>v</b>
<b>Índice de Tabelas</b> .....	<b>vii</b>
<b>Resumo</b> .....	<b>viii</b>
<b>Abstract</b> .....	<b>ix</b>
<b>1. Introdução</b> .....	<b>1</b>
1.1 Enquadramento e Motivação.....	1
1.2 Problema e Objetivo.....	1
1.3 Abordagem Metodológica .....	1
1.4 Estrutura do Projeto.....	3
<b>2. Revisão da Literatura</b> .....	<b>4</b>
2.1 Sistemas de Informação .....	4
2.2 Integração de Sistemas de Informação.....	6
2.3 ERP (Enterprise Resource Planning) .....	11
2.4 Redes Sociais .....	13
2.5 CRM (Customer Relationship Management) .....	13
2.6 Relação entre ERP, CRM e Redes Sociais .....	14
2.7 Síntese .....	16
<b>3. Modelo Conceptual</b> .....	<b>19</b>
3.1 A Arquitetura do Sistema .....	19
3.2 Funcionalidades do sistema .....	21
3.3 Modelo de Classes.....	22
3.4 Descrição das Funcionalidades .....	24
<b>4. Desenvolvimento do Protótipo</b> .....	<b>30</b>
4.1 Introdução .....	30
4.2 Requisitos .....	30
4.3 Desenvolvimento .....	31

<b>5. Testes funcionais e avaliação preliminar .....</b>	<b>42</b>
5.1 Funcionalidade Login.....	42
5.2 Funcionalidade Criar Campanha .....	44
5.3 Funcionalidade Editar Campanha .....	48
5.4 Funcionalidade Remover Campanha .....	50
5.5 Funcionalidade Publicar Campanha.....	53
5.6 Análise e utilização .....	55
<b>6. Conclusões .....</b>	<b>58</b>
6.1 Trabalho Futuro.....	59
<b>7. Bibliografia .....</b>	<b>60</b>
<b>8. Anexos .....</b>	<b>63</b>
Anexo 1 - Código Fonte .....	64

# Índice de Figuras

Figura 1: Framework - Fonte: March & Smith (1995) .....	2
Figura 2: Áreas de funcionamento de um SI - Fonte: Stair & Reynolds (2013).....	4
Figura 3: Fragmentação Vertical das unidades de negócio (Adaptado de Hasselbring, 2000).....	6
Figura 4: Horizontal para suportar processos de negócio (Adaptado Hasselbring, 2000).....	7
Figura 5: Componentes da arquitetura de aplicação (Adaptado Chari & Seshadri, 2004) .....	8
Figura 6: Componentes dos níveis de integração (Adaptado Chari & Seshadri, 2004) .....	9
Figura 7: Componentes da especificidade dos standards da industria - Fonte: Chari, K.& Seshadri, S. (2004).....	10
Figura 8: Quota de Mercado ERP - Fonte: Forbes.com (2013) .....	12
Figura 9: Principais aplicações de CRM Operacional - Fonte: Buttle (2008).....	14
Figura 10: Revisão dos temas da literatura (parte 1).....	18
Figura 11: Revisão dos temas da literatura (parte 2).....	18
Figura 12: Demonstração do sistema.....	19
Figura 13: Use Case do Sistema .....	21
Figura 14: Diagrama de Classes.....	22
Figura 15: Diagrama de Atividades Login.....	24
Figura 16: Diagrama de Atividades Criar Campanha.....	25
Figura 17: Diagrama de Atividades Editar Campanha.....	26
Figura 18: Diagrama de Atividades Remover Campanha.....	27
Figura 19: Diagrama de Atividade Publicar Campanha.....	28
Figura 20: Sistema de Autenticação.....	33
Figura 21: Função responsável pelo load da página de autenticação .....	34
Figura 22: Menu Principal .....	34
Figura 23: Formulário de criação de campanha.....	35
Figura 24: Sistema de escolha múltipla.....	36
Figura 25: Preenchimento das caixas de seleção.....	36
Figura 26: Sistemas de exceções de artigos .....	37
Figura 27: Adição da campanha .....	37
Figura 28: Edição de uma campanha .....	38
Figura 29: Código edição de uma campanha .....	38
Figura 30: Algoritmo de pesquisa.....	39
Figura 31: Formulário publicar .....	40
Figura 32: Código responsável por publicar a campanha .....	40

Figura 33: Publicar Estado.....	41
Figura 34: Código Publicar Estado.....	41
Figura 35: Tutorial de página login.....	43
Figura 36: Tutorial introdução dados.....	43
Figura 37: Tutorial menu principal.....	44
Figura 38: Tutorial menu principal 2.....	45
Figura 39: Tutorial menu criar campanha.....	46
Figura 40: Tutorial menu criar campanhas preenchido.....	46
Figura 41: Tutorial adicionar e remover artigos.....	47
Figura 42: Tutorial campanha adicionada.....	47
Figura 43: Tutorial menu principal 3.....	48
Figura 44: Tutorial seleção editar campanha.....	49
Figura 45: Tutorial dados editar campanha.....	49
Figura 46: Tutorial edição campanha completa.....	50
Figura 47: Tutorial menu principal 4.....	51
Figura 48: Tutorial seleção de campanha a remover.....	51
Figura 49: Tutorial remoção completa.....	52
Figura 50: Tutorial remoção completa.....	52
Figura 51: Tutorial menu principal 5.....	53
Figura 52: Tutorial listagem publicação campanhas.....	54
Figura 53: Tutorial publicado com sucesso.....	54
Figura 54: Tutorial output da publicação.....	55
Figura 55: Analise SWOT.....	56

## Índice de Tabelas

Tabela 1: Tipos de SIs nas organizações - Fonte: Rainer & Cegielski (2010).....	5
Tabela 2: Fases da implementação de um ERP - Fonte: Granta, Hwanga, Tuc (2013) .....	11
Tabela 3: Tabela das Campanhas. ....	32
Tabela 4: Tabela dos Grupos Clientes. ....	32
Tabela 5: Tabela dos Grupos Artigos.....	32
Tabela 6: Tabela dos Artigos a não publicar. ....	32
Tabela 7: Resultado dos testes.....	55

## **Resumo**

O presente trabalho apresenta uma solução conceptual para a problemática de integração de ERP com Redes Sociais. Para isto é proposto um modelo conceptual de integração entre estes sistemas. Baseado no modelo proposto foi implementada uma aplicação que permite a integração entre ERP e redes sociais usando CRM.

O ERP utilizado nesta implementação foi o SAP. A escolha do ERP baseou-se com o facto de tentar criar uma solução o mais abrangente possível, já que este é o ERP com mais quota de mercado atualmente. Para rede social foi seleccionada o Facebook por motivos semelhantes de mercado.

Foi efetuada uma revisão da literatura, que inclui uma preocupação com todos os temas necessários e relevantes para a resolução e compreensão deste problema.

A aplicação implementada consiste em criar campanhas publicitarias utilizando dados diretamente do ERP em questão e exportando-os de forma especifica, controlada e personalizada para o Facebook. Os utilizadores desta aplicação podem personalizar aspetos das campanhas publicitarias como os artigos pertencentes, os clientes que vão receber a campanha entre outros aspetos.

O modelo desenvolvido é facilmente aplicado a uma qualquer industria e pode servir como suporte para empresas informáticas que pretendam desenvolver software nesta área.

**Palavras Chave:** CRM, ERP, SAP, Facebook

**Classificação:** JEL: M15 - IT Gestão, JEL: M30 - Geral



## **Abstract**

The present work presents a conceptual solution to solve the problems related with a ERP/Social Network integration. In order to fulfill that goal a conceptual model was created. Based on this model an application was proposed that allows such integration.

The ERP system used in this implementation was SAP. The choice of this ERP was based on the fact that we are trying to create the broadest solution possible since SAP is currently the ERP with more market share. The social network selected for this work was Facebook, it was chosen for the same reasons than SAP.

A literature review was carried out, which includes a concern with every theme necessary and relevant to the resolution and understanding of this problem.

The application consists on creating publicity campaigns using data from the ERP database and exporting it, in a specific, controlled, and personalized way to Facebook. The users of this application will be able to personalize the aspects of publicity campaigns they create such as items or the clients that will receive that particular campaign.

The model developed is easily applied to any industry and it can serve as support for other IT companies that intend to develop software on this field.

**Key Words:** CRM, ERP, SAP, Facebook

**JEL Classification:** M15 - IT Management, M30 - General

# **1. Introdução**

## **1.1 Enquadramento e Motivação**

Com a 'explosão' mundial e rápida dos media sociais, as empresas estão sobre uma grande pressão para se conectarem com os seus clientes. Cada vez mais a atividade destes clientes aumenta virtualmente, localizada em sites de media social e redes sociais. A velocidade de adoção destas novas tecnologias é elevada considerando quer consumidores quer empresas. Em 2010 existiam mais de 500 milhões de utilizadores ativos no Facebook, e embora esta rede social tenha sido originalmente criada no Estados Unidos, mais de 70% dos seus utilizadores encontra-se fora desse mesmo país. Em Março 2010, mais de 10 mil milhões de tweets tinham sido publicados pelo o Twitter, rede social criada apenas em 2006. Em Julho desse mesmo ano, e apenas quatro meses depois, o numero de tweets tinha duplicado para 20 mil milhões. Estes dados indicam o local onde os consumidores estão concentrados e onde as empresas atualmente devem estar. As redes sociais detêm um enorme potencial para as empresas se conectarem com os seus clientes e, por consequência desse processo, facilitar a entrada de receitas, e reduzir custos. O projeto desenvolvido neste documento está inserido num contexto empresarial uma vez que foi proposto por uma empresa e provém de uma necessidade real.

## **1.2 Problema e Objetivo**

A problemática neste projeto consiste na integração das redes sociais com sistemas de ERP, sendo este um problema específico na área de sistemas de informação designada por integração de sistemas. Este projeto tem como objetivos:

- A criação de uma proposta de um modelo conceptual que permita a integração de ERP com redes sociais.
- A validação desse modelo através da criação de um protótipo

## **1.3 Abordagem Metodológica**

A abordagem metodológica em tecnologias de informação segundo March & Smith (1995) pode ser dividida em duas partes. A primeira denominada de 'natural science' é definida pelos autores como um método de pesquisa mais tradicional que se centra maioritariamente em domínios físicos, biológicos e sociais. O seu principal produto é a

criação de teorias. Este tipo de abordagem segundo Kaplan (1964) é constituído por norma por duas atividades: descobrimento e justificação. No entanto tendo em conta os objetivos acima propostos para este projeto, este tipo de abordagem não é o ideal.

A segunda abordagem ou 'design science' é definida por March & Smith (1995) como uma tentativa de criar coisas com propósito humano. As duas principais atividades são construir e avaliar. Estes autores propõem uma framework baseada em design science (Figura 1).

ACTIVIDADES DE PESQUISA				
	Construir	Avaliar	Teorizar	Justificar
OUTPUT DE PESQUISA				
Conceitos				
Modelo				
Metodos				
Instanciação				

Figura 1: Framework - Fonte: March & Smith (1995)

No output de pesquisa os conceitos constituem a conceptualização utilizada para descrever problemas. Os modelos são um conjunto de preposições ou expressões que expressam a relação entre os conceitos. Os métodos são algoritmos utilizados para executar tarefas e a instanciação consiste em operacionalizar os conceitos, modelos e métodos.

Como é possível ver na Figura 1 a framework proposta produz dezasseis células para investigação. No entanto nem todas necessitam de estar preenchidas num modelo de pesquisa.

Neste projeto serão utilizadas as células de interseção das linhas de 'Conceitos', 'Modelo' e 'Instanciação' com as colunas 'Construir' e 'Avaliar'. Cada célula contem um objetivo específico que é apresentado e explicado em cada capítulo deste documento. A interseção da coluna construir com a linha de conceitos foi resolvida efetuando

pesquisa, isto é, uma revisão da literatura sobre a problemática deste projeto. A interseção com o modelo resultou na resolução de um dos objetivos deste projeto, a criação de um modelo conceptual. A última intersecção da coluna construir foi com a linha da instanciação que resolveu o segundo objetivo deste projeto, a criação de um protótipo para verificar a exequibilidade do modelo proposto.

As atividades na coluna de avaliação serviram para avaliar se todo o processo de construção foi efetuado corretamente. Na revisão da literatura, foi necessário garantir que todos os tópicos tinham sido abordados por vários autores no modelo, e, como este projeto provem de uma necessidade real, foi necessário garantir que todos os detalhes estavam alinhados com essa necessidade e no protótipo foram efetuados testes para garantir a sua funcionalidade.

#### **1.4 Estrutura do Projeto**

No primeiro capítulo é realizado o enquadramento do projeto com realidade atual. É referida a problemática, são definidos os objetivos e é abordada a metodologia a ser seguida. No Capítulo 2 é apresentada a revisão da literatura onde se realiza o enquadramento teórico, a identificação de soluções propostas, e onde são identificadas pistas para a proposta do modelo. No Capítulo 3 é descrito e proposto o modelo conceptual usando como ferramenta principal UML. O Capítulo 4 apresenta o desenvolvimento e implementação do protótipo baseado no modelo conceptual. O Capítulo 5 demonstra os testes efetuados na solução e o Capítulo 6 apresenta as conclusões finais e trabalho futuro concluindo assim o projeto.

## 2. Revisão da Literatura

### 2.1 Sistemas de Informação

Um sistema de informação pode ser definido como um conjunto de componentes inter-relacionados que colecionam ou recuperam, processam, armazenam e distribuem informação para suportar o processo de decisão numa organização. Em adição ao suporte de decisão, coordenação e controlo, os sistemas de informação podem ajudar os gestores e os trabalhadores e analisar problemas, visualizar temas complexos e criar novos produtos (Laudon & Laudon 2006).

Como referido anteriormente um sistema de informação pode ser definido como um conjunto de componentes. Este conjunto segundo Stair & Reynolds (2013) é constituído por quatro áreas: Input, Processamento, Output e Feedback (Fig. 2).

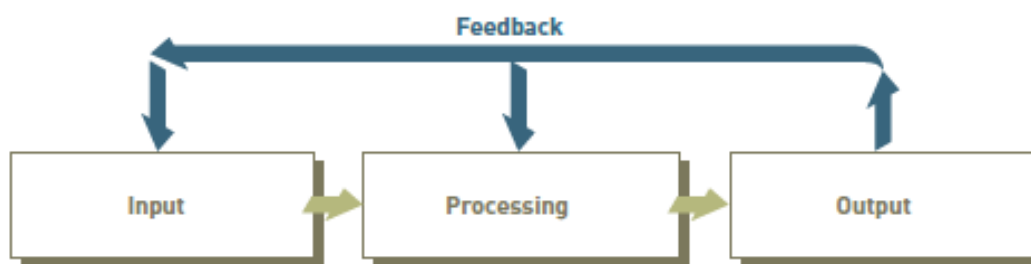


Figura 2: Áreas de funcionamento de um SI - Fonte: Stair & Reynolds (2013)

O input é onde existe captura de dados brutos ou seja que ainda não foram processados. Esses dados brutos seguem de seguida para a fase de processamento onde são convertidos e transformados em dados úteis. Este processo pode ser realizado através de cálculos, comparação de dados, ações específicas relacionadas com o negócio ou ainda podem ser armazenados para utilização futura. O processamento destes dados em informação é crítico para o funcionamento eficaz do sistema. A fase de Output é normalmente caracterizada pelo o fornecimentos de informação útil, sendo o mais comum esta ser providenciada em forma de documentos e relatórios, e, em certos casos, o Output de um sistema pode ainda representar um Input para outro sistema. Por último a fase de Feedback é a informação que o sistema transmite que permite modificar os Inputs ou as ações na fase de processamento de acordo com as necessidades do negócio.

Embora possuindo as mesmas áreas gerais cada empresa emprega um tipo diferente de sistema de informação dependendo da área do negócio ou necessidades específicas.

Segundo Rainer & Cegielski (2010) os sistemas de informação podem ser divididos em 10 tipos, cada um com funções diferentes. A Tabela 1 demonstra os vários tipos de sistemas de informação e as suas funções distintas.

**Tabela 1: Tipos de Sistemas nas organizações (adaptado de Rainer & Cegielski, 2010)**

<b>Tipo de Sistema</b>	<b>Função</b>	<b>Exemplo</b>
SI por área funcional	Suporta as atividades dentro de uma área específica.	Sistema de processamento de ordenados
Sistema de processamento de transações	Processa dados de transações de eventos de negócio.	Terminais de ponto de vendas do Walmart
ERP	Integra todas as áreas funcionais da organização.	Oracle, SAP
Sistema automatizado de escritório	Suporta atividades diárias de indivíduos ou grupos.	Microsoft Office
Sistemas de informação de gestão	Produz relatórios de dados de transações normalmente relativos a uma área em particular.	Relatórios do total de vendas por cada cliente
Sistemas de suporte à decisão	Fornecer acesso a dados e a ferramentas de análise.	Analisar mudanças no orçamento
Sistema 'Expert'	Imita um ser humano numa determinada área nas decisões a tomar.	Análise de aprovação de cartões de crédito
'dashboard' executivo	Fornecer informação estruturada e resumida sobre todos os aspetos relacionados com o negócio para gestores de topo.	Estado de produtos em vendas
Sistema de cadeia de abastecimento	Gere o fluxo de materiais, serviços e informação entre organizações.	'Walmart Retail Link' que cria uma ligação do Walmart aos fornecedores
Sistemas de comércio eletrónico	Possibilita transações entre organizações e entre organizações e clientes.	www.dell.com

## 2.2 Integração de Sistemas de Informação

A integração deste tipo de sistemas é complexa e segundo Hasselbring (2000) para uma integração de um sistema de informação com sucesso deve-se dividir as unidades organizacionais em 3 camadas (Fig.3): arquitetura de negócio, arquitetura aplicacional e arquitetura de tecnologia. A arquitetura do negócio define a organização da estrutura da empresa e os workflows dos processos de negócio. A arquitetura aplicacional define a implementação dos conceitos de negócio em relação às aplicações necessárias, e por fim, a arquitetura de tecnologia define a infraestrutura da comunicação e informação.

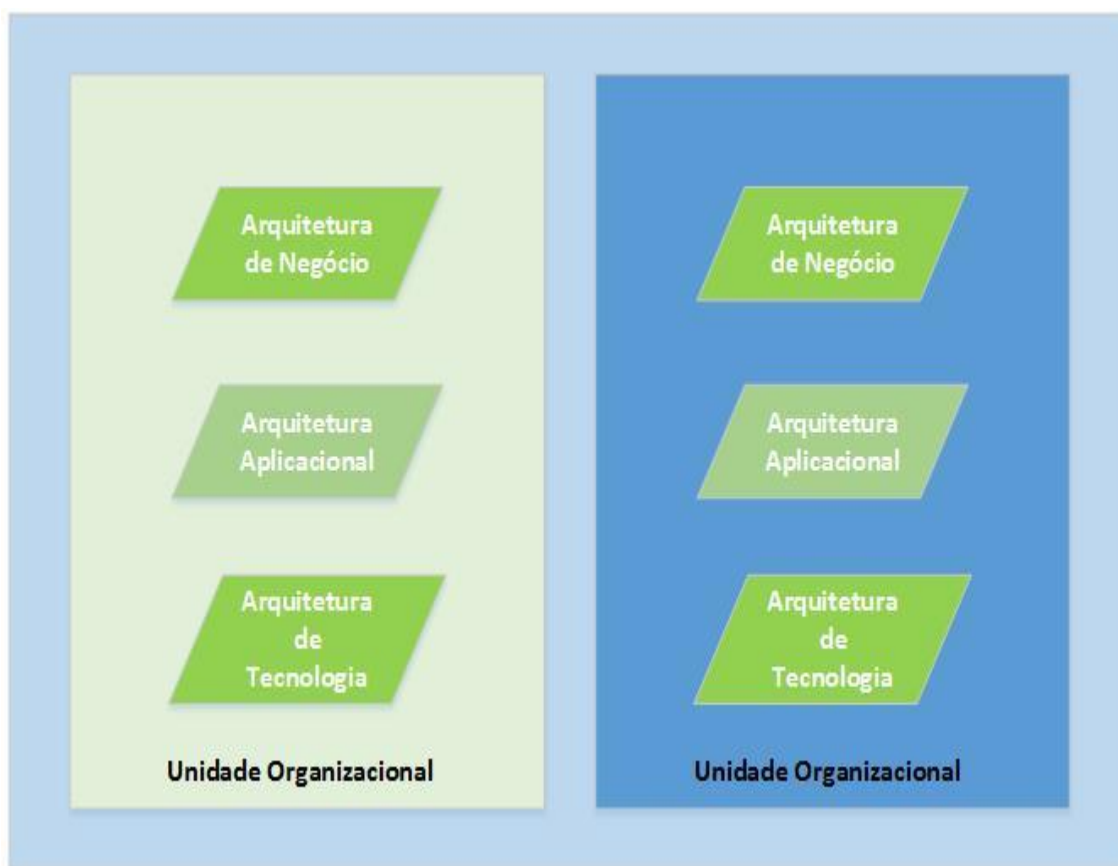


Figura 3: Fragmentação Vertical das unidades de negócio (Adaptado de Hasselbring, 2000)

No entanto esta integração não pode apenas depender de uma divisão vertical, mas também tem que consistir numa integração horizontal de forma a suportar os processos de negócio (Fig. 4). Esta integração é composta por 3 níveis: processos inter-organizacionais, onde os engenheiros de negócio procuram que todos os processos façam parte da estrutura horizontal da empresa; integração das aplicações das empresas, onde o objetivo passa por integrar sistemas de ERP (Enterprise resource planning); integração de Middleware. Neste nível são utilizadas técnicas para construir

infraestruturas topo de gama. A junção destas duas fases permite uma integração de sistemas de informação com sucesso. Um exemplo de sistemas informação é chamado de Enterprise resource planning (ERP).

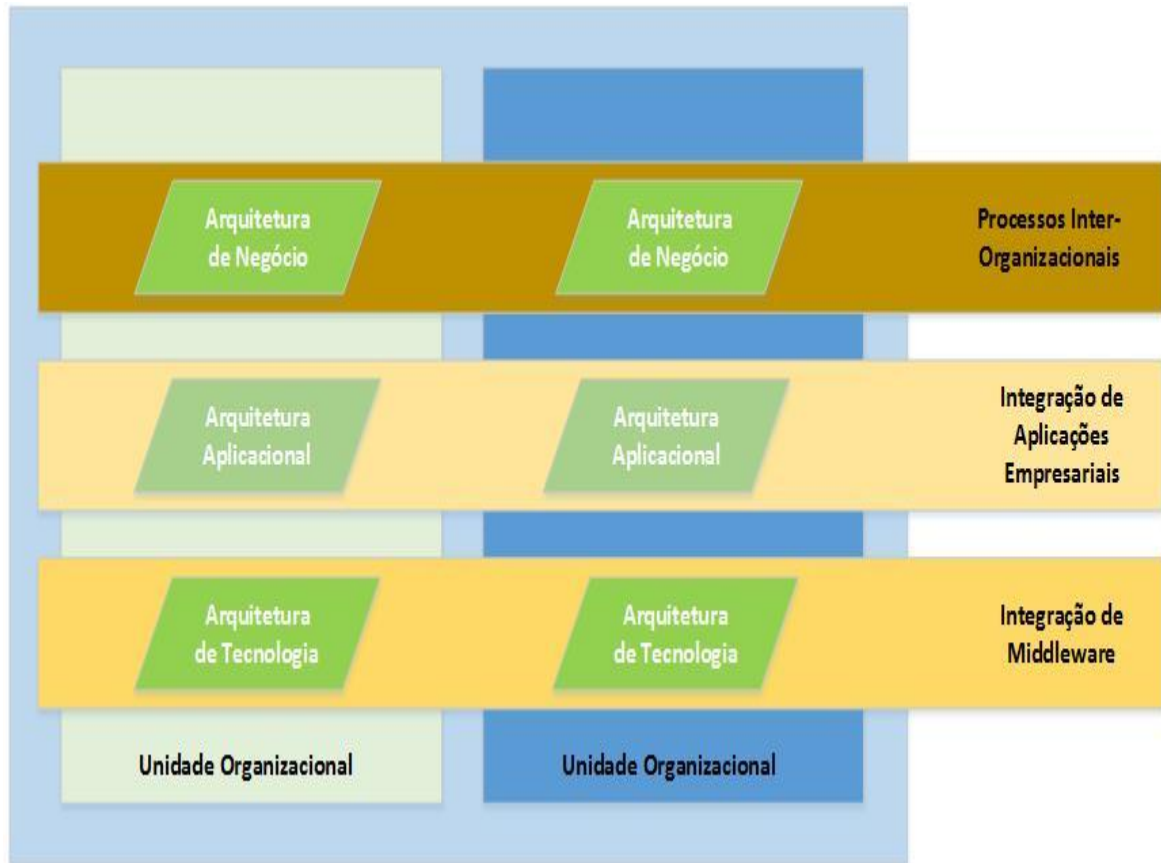


Figura 4: Horizontal para suportar processos de negócio (Adaptado Hasselbring, 2000)

No entanto esta não é a única possibilidade de integração de Sistemas de Informação. Chari & Seshadri (2004) propõem uma Framework com as fundações e perspetivas necessárias, ajudando a identificar problemas que são muitas vezes ignorados pelos standards atuais de integração de sistemas.

Esta Framework é dividida em três dimensões: ‘Arquitetura Aplicacional’, ‘Nível de Integração’, e ‘Especificidade dos Standards Industria’.

A primeira dimensão ‘Arquitetura Aplicacional’ (Fig. 5) possui três componentes principais. O primeiro componente é a apresentação responsável pelas interações entre o sistema e as suas interfaces. O segundo trata da lógica de negócio, onde são implementadas todas as regras específicas do negócio em questão. A terceira tem a ver com a lógica de dados para poderem ser acedidos e compreendidos pela lógica de negócio.



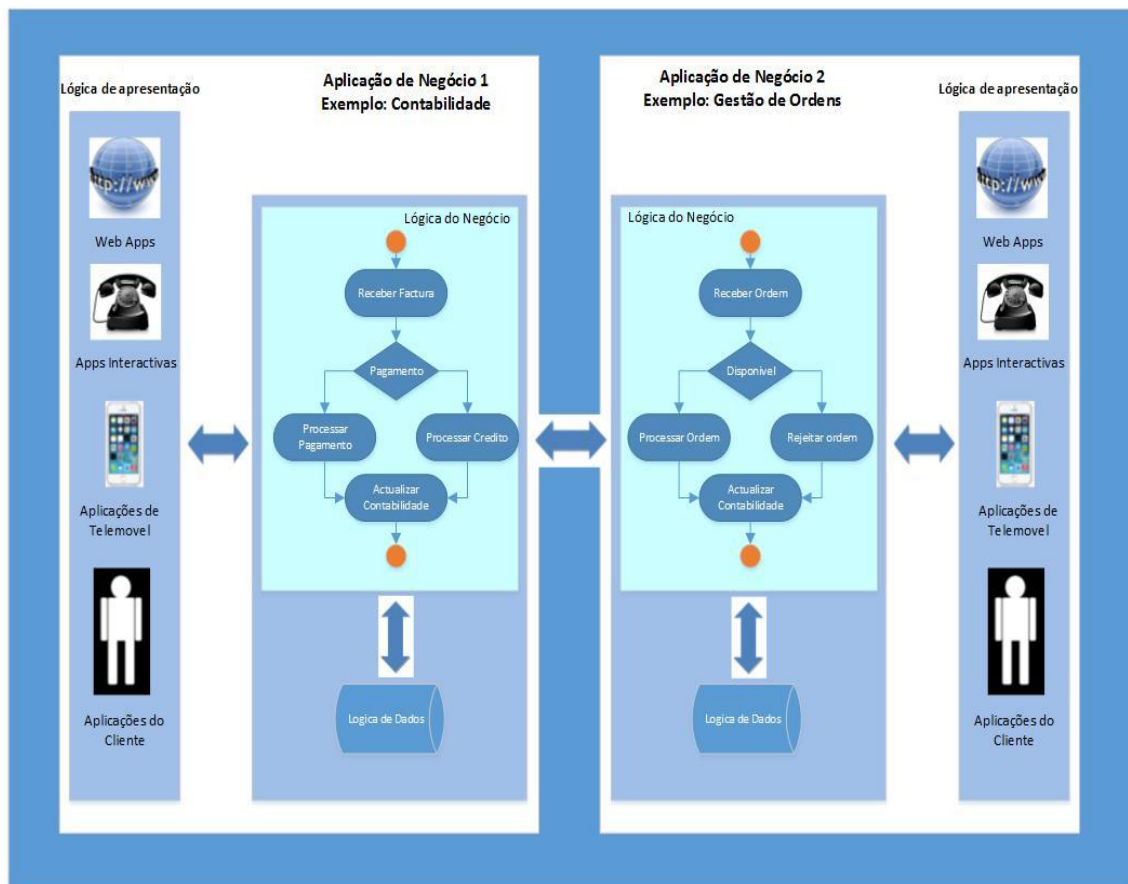


Figura 5: Componentes da arquitetura de aplicação (Adaptado Chari & Seshadri, 2004)

A segunda dimensão ‘Nível de Integração’ (Fig. 6) representa os níveis de integração possíveis sendo que existem três: Transporte, Dados, e Processos. O primeiro nível de integração 'Transporte' fornece infra-estrutura para transportar dados entre sistemas semelhantes e não semelhantes. O segundo nível de integração ‘Dados’ facilita a integração de várias aplicações através da representação de dados. O último nível de integração ‘Processo’ refere-se a integração de, pelo menos duas entidades diferentes como, por exemplo aplicações e linhas de negócio.

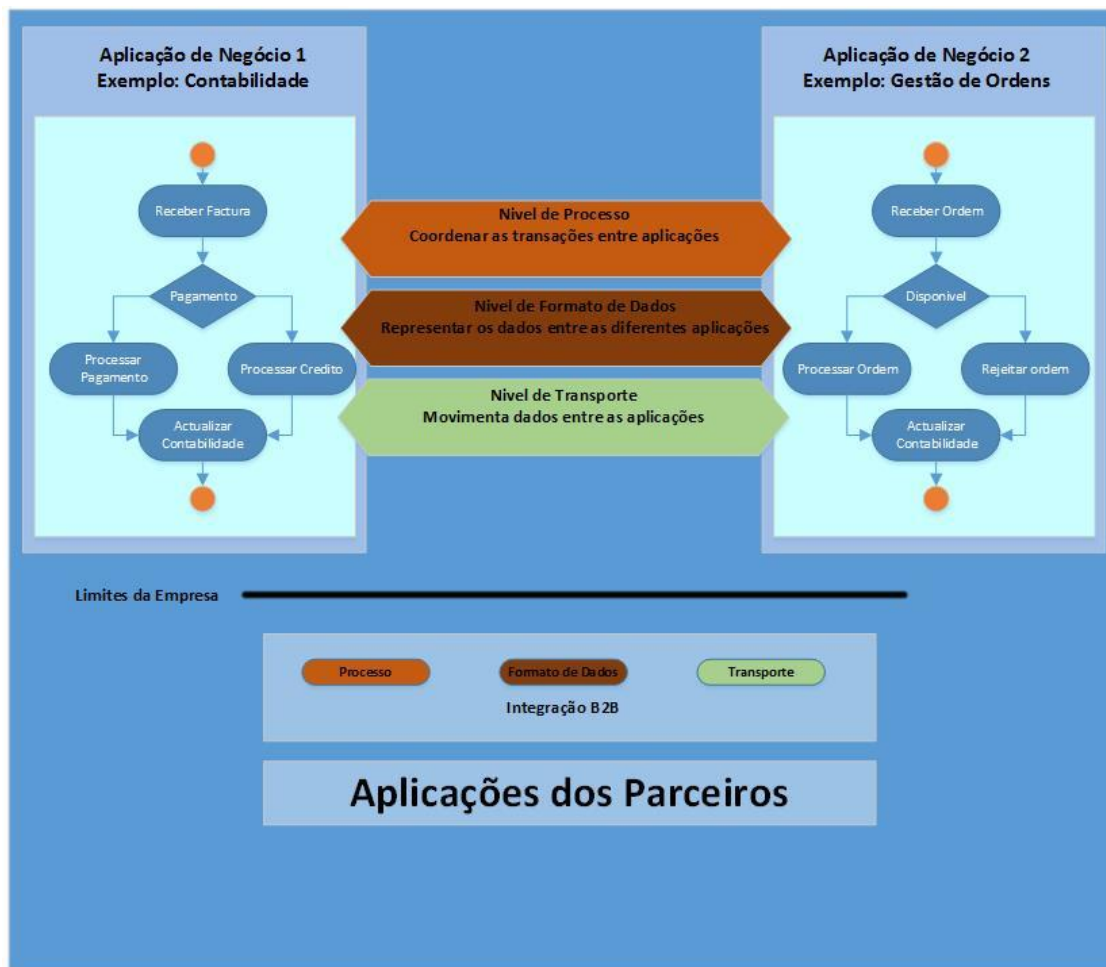


Figura 6: Componentes dos níveis de integração (Adaptado Chari & Seshadri, 2004)

A terceira dimensão da Framework 'Especificidade dos Standards Industria' (Fig. 7) possui dois componentes, e é referente ao âmbito dos standards de integração, i .e. se são genéricos e como tal aplicáveis a uma variedade de indústrias, ou específicos e portanto, fechados a uma indústria particular. O primeiro componente é a Independência da Indústria, onde os standards são genéricos como HTTP e SQL. O segundo componente é específico a uma indústria como exemplo RossetaNet.

Industry Domain	Applications Architecture	Integration Level	Standard	Industry Domain	Applications Architecture	Integration Level	Standard		
Domain Independent	Data Logic	Transport	FTP	Domain Dependent	Data Logic	Transport	ACORD XML Business Message Specification for P&C Insurance and Surety 1.4.1		
			OSI-RPC				RosettaNet Implementation Framework 2.0		
			SMTP				ANSI X12N		
			SQL/CLI				HL7 2.5		
			HTTP 1.1				HL7 CCOW 1.3		
			CORBA/IOP 3.0.2				ISO 15022		
			ebXML-Message Service 1.0				ACORD Automation Level 3		
			OAMAS 1.0				ACORD OLiE 2.6.00		
			JDBC 3.0				ACORD XML Business Message Specification for P&C Insurance and Surety 1.4.1		
			JMS 1.1				RosettaNet Dictionaries		
		ODBC 3.51	RosettaNet Implementation Framework 2.0						
		WS-Routing	SPEC2000						
		Data Format	SQL			Process	HL7 2.5		
			MIME				HL7 CCOW 1.3		
			S/MIME 3.0				RosettaNet Partner Interface Process		
			ebXML-Message Service 1.0				SPEC2000		
			ebXML Core Components (Document Assembly) 1.04				Business Logic	Transport	ACORD XML Business Message Specification for P&C Insurance and Surety 1.4.1
			OAGIS-BOD 8.0						RosettaNet Implementation Framework 2.0
			SOAP 1.2					Data Format	ANSI X12N
			XML-DTD 1.1						HL7 2.5
	XML Schema		HL7 2.5						
	XQuery 1.0		ISO 15022						
	JDBC 3.0	ACORD Automation Level 3							
	ODBC 3.51	ACORD OLiE 2.6.00							
	OAGIS-BOD 8.0	ACORD XML Business Message Specification for P&C Insurance and Surety 1.4.1							
	UML 1.5	RosettaNet Dictionaries							
	WSDL 2.0	RosettaNet Implementation Framework 2.0							
	SPEC2000								
	Process	Business Logic	Transport		ebXML-CPP 2.0				
					OAMAS 1.0				
					EDIFACT				
					ANSI X.12				
					X.12				
					ebXML Core Components (Document Assembly) 1.04				
					OAGIS-BOD 8.0				
					SOAP 1.2				
					WFMC Interfaces				
					BizTalk Framework 2.0				
	Presentation Logic	Business Logic	Data Format		ebXML-BPSS 1.01				
					ebXML-CPA 2.0				
					OAGIS-BOD 8.0				
					OAGIS-Integration Scenarios 8.0				
					UML 1.5				
					WFMC Reference Model and Interfaces				
					BPML 1.0				
					BizTalk Framework 2.0				
					BPEL4WS 1.1				
ISO/IEC 6429:1992									
Presentation Logic	Presentation Logic	Transport	Telnet						
			HTTP 1.1						
			TN 3270E						
			WAP 2.0 (lower layers)						
			WAP 2.0 (higher layers)						
		Data Format	HTML 4.01						
			XSL 1.0						
			WAP-WML 2.0						
			JSP 2.0						
			UML 1.5						
Presentation Logic	Presentation Logic	Process	HL7 CCOW 1.3						
			HL7 CCOW 1.3						
			HL7 CCOW 1.3						

Figura 7: Componentes da especificidade dos standards da industria - Fonte: Chari, K.& Seshadri, S. (2004)

A integração destes sistemas varia de área para área e de tipo de sistemas para tipo de sistema. Num exemplo de uma integração mais específica, e com os serviços web a cada vez mais emergirem na integração de sistemas de informação como tecnologia de suporte para aplicações autónomas através de protocolos de internet, as frameworks de software para integrações de sistemas de informação de acordo com Costa, Resende, Nakamura, & Segatto, (2008), devem também especificar soluções parciais para a sua integração.

Noutro caso específico de integração de sistemas de informação, as unidades de saúde cada vez mais dependem de sistemas de informação que tratem eficazmente do registo do paciente, admissão, dar alta, entre outras funções médicas, administrativas e financeiras. Aqui tal como noutras áreas de integração de Sis, certos problemas

específicos são levantados e várias abordagens podem ser tomadas para a sua resolução. Estas abordagens segundo Sabooniha, Toohey, & Lee, (2012) podem ser: Orientadas a mensagens, a aplicações, a coordenação ou orientadas a middleware.

Para outros casos específicos como sistemas de integração de contabilidade existe ainda frameworks como a ‘REA accounting framework’ entre outras (Schwaiger & Abmayer, 2013).

### 2.3 ERP (Enterprise Resource Planning)

O Enterprise resource planning ou ERP é um sistema de informação que possibilita um agrupamento e a interligação de vários módulos numa organização empresarial, tendo em comum a sua base de dados (Pinto 2010). Este sistema pode ainda ser considerado um conjunto de técnicas e conceitos para uma gestão integrada de negócios como um todo do ponto de vista do uso efetivo de recursos de gestão para melhorar e eficiência da gestão empresarial (Leon 2008). A implementação deste tipo de sistemas pode ser realizada através de 6 fases segundo Granta, Hwanga, Tuc (2013)

**Tabela 2: Fases da implementação de um ERP - (Adaptado de Granta, Hwanga, Tuc, 2013)**

Fase	Descrição
Integração especificação-sistema	É a fase mais baixa dos níveis de integração e a sua preocupação principal é com a integração específica e compatibilidade. A integração específica foca-se nas especificações do software, hardware e aplicações para equipamento móvel.
Integração utilizador-sistema	Nesta fase a preocupação principal é assegurar que os utilizadores sejam integrados com a tecnologia e o ambiente. Envolve dois tipos de integração: ergonómica e cognitiva. A ergonómica assegura que o sistema e o ambiente são desenhados de forma ergonómica isto significa a interface gráfica, teclado, software e hardware têm que ser user-friendly. A integração cognitiva assegura que as comunicações entre o sistema e o utilizador são úteis e consistentes.
Ilhas de integração tecnológica	O objetivo é ligar “ilhas” de tecnologia dispersadas pela empresa. A integração a este nível preocupa-se com a capacidade de comunicações entre estas ilhas.
Integração da organização	Preocupa-se com a integração que gere os esforços das várias funções ao longo dessa cadeia de valor. Envolve quatro tipos de integração: Integração interna vertical, horizontal, estratégica e temporal interna. Integração vertical consiste na passagem de informação da gestão estratégica para trabalhadores e vice-versa. Integração estratégica mede quão bem os sistemas de informação que suportam a organização, e a integração temporal interna mede a efetividade e coordenação que existe entre grupos, funções, departamentos e indivíduos.

Integração socio-organizacional	Envolve a ligação entre a empresa e os clientes, fornecedores, parceiros estratégicos, governo, e instituições cívicas. Integra a gestão de relação com os clientes e a gestão da cadeia logística
Integração Global	Empresas devem operar como entidades globais em vez de geograficamente e devem ser vistas como internacionais com componentes domésticos. A fase 6 de integração preocupa-se com a integração ao longo de nacionalidades e culturas.

O ERP com maior base instalada é o SAP, com cerca de 25% da quota de mercado (Fig. 8).

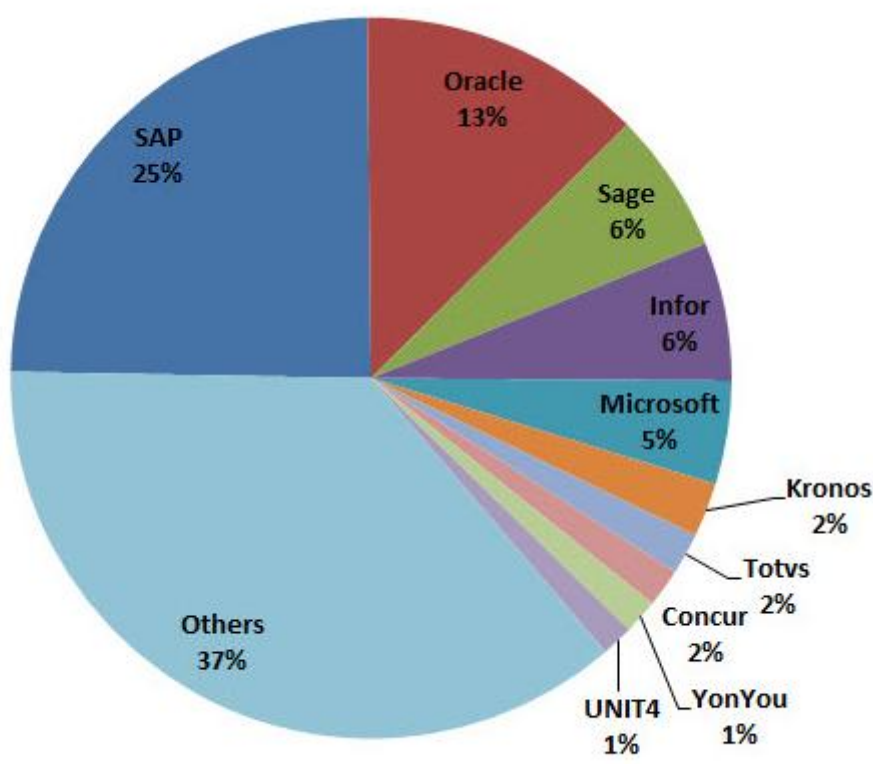


Figura 8: Quota de Mercado ERP - Fonte: Forbes.com (2013)

De acordo com o site do SAP ([www.sap.com](http://www.sap.com)) a empresa tem sede em Walldorf na Alemanha, é a maior fornecedora de aplicações corporativas e umas das maiores empresas de software a nível mundial. Foi fundada em 1972 por um grupo de ex-engenheiros da IBM com uma visão única: combinar todas as funções de um negócio numa só aplicação, permitindo ainda, a criação de addons que correm dentro da aplicação ‘mãe’ podendo ter como objetivo a criação de funcionalidades nos processos de negócio até à ligação dos mesmos a redes sociais.

## **2.4 Redes Sociais**

Redes sociais de acordo com Boyd & Ellison (2007), podem ser definidas como serviços web que possibilitam que pessoas (utilizadores) construam perfis que podem ser públicos ou semipúblicos dependendo das restrições impostas, permitindo ainda definir a lista de pessoas com quem pretendem interagir, ver e cruzar todas as conexões criadas por outros dentro desse sistema. A natureza destas conexões varia de site para site. Devido a grande utilização nestes sites a publicidade tem vindo a aumentar os mesmos de forma progressiva.

Lepoint (2012) defende que atualmente os 'Marketers' estão obcecados com as redes sociais, especialmente com o Facebook e com uma métrica em particular: "o valor de um seguidor". Esta métrica faz com que existia uma grande pressão nas pessoas ligadas ao Marketing para conseguir mais, com menos recursos financeiros. O Facebook e outras redes sociais segundo Bolotaeva & Cata (2010), oferecem uma percentagem de retorno alta permitindo chegar a muito público sem utilizar grandes recursos financeiros.

## **2.5 CRM (Customer Relationship Management)**

O CRM pode ser definido segundo Buttle (2008) como uma estratégia central de negócio que integra processos internos, funções e redes externas, para criar valor em consumidores chave, retornando lucro. As suas bases são a grande qualidade de dados relacionados com o consumidor e a utilização das tecnologias de informação. Existem três tipos de CRM.

CRM Estratégico é focado no desenvolvimento de uma cultura centrada no cliente. Esta cultura é focada em ganhar e manter clientes ao criar valor melhor que os seus concorrentes (Kumar. 2010). De modo igual reflete-se no comportamento das lideranças, no design dos sistemas da companhia e nos mitos e histórias criadas dentro da empresa.

O CRM operacional automatiza e melhora a comunicação com o cliente em pessoa e o apoio a ser prestado. O software aplicativo permite que marketing, vendas, e outros serviços sejam totalmente integrados e automatizados. Segundo Buttle (2008) a Figura 9 demonstra algumas das maiores aplicações de CRM operacional.

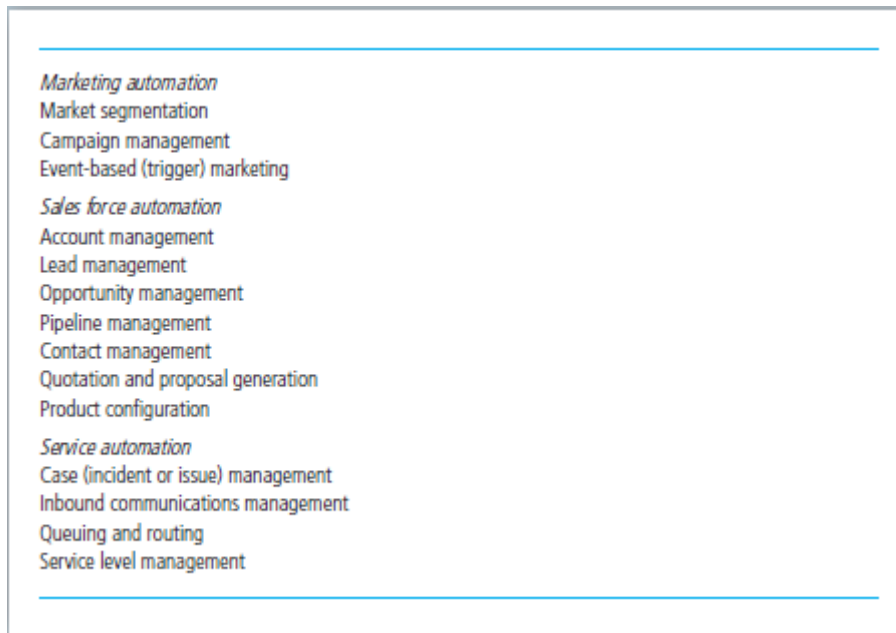


Figura 9: Principais aplicações de CRM Operacional - Fonte: Buttle (2008)

O CRM Analítico centra-se em capturar, armazenar, extrair, integrar, processar, interpretar, distribuir, usar e gerar relatórios de dados relacionados com os clientes para melhorar quer o valor do cliente, quer o da empresa. A base deste tipo de CRM é a informação do cliente. Esta informação pode estar localizada na base de dados da empresa, e a estes dados de ordem interna podem ainda ser acrescentados os de ordem externa. Através das análises destes dados é possível ainda responder a outras questões tais como: Quais os clientes mais valiosos, quais tem mais propensão a mudar para os concorrentes ou quais vão responder a uma oferta particular?

Por último, outro conceito chave nesta área é CRM Social. É definido por Woodcock, Neil, Green, & Michael Starkey (2011) como uma estratégia de negócio de interação com os clientes através de Social Media com o objectivo de criar confiança e lealdade à marca.

## 2.6 Relação entre ERP, CRM e Redes Sociais

O ERP ou Enterprise Resource Planning e o CRM ou Customer Relationship Management são ambos sistemas desenhados para automatizar e melhorar os processos de negócio. Enquanto o CRM é um sistema para gerir a imagem do negócio, o ERP gere os sistemas na retaguarda do negócio. A relação entre estes sistemas é necessária para trabalharem em conjunto um com o outro de modo que ambos funcionem de forma eficiente e eficazmente (Lorette 2014).

O CRM é o processo de como uma empresa gere os seus contactos. Como o nome implica, o CRM foca-se no como, quando, onde e como é que se pode interagir com os seus clientes (potenciais e atuais). O ERP é um sistema de informação que possibilita um agrupamento e a interligação de vários módulos numa organização empresarial, tendo em comum a sua base de dados. Este sistema foca-se em automatizar processos como distribuição, contabilidade, vendas, produção entre outros. (Barr 2013)

Como referido previamente, o ERP e o CRM, focam-se em diferentes partes da organização, e segundo Lorette (2014) um é a 'cara' da organização e o outro gere as operações internas da empresa. No entanto estes sistemas podem se cruzar. Alguns sistemas de ERP já incluem certas funções de CRM, da mesma forma que alguns softwares de CRM incluem funcionalidades de ERP. Embora a maior parte dos vendedores destes softwares os vendam em separado, outros já estão a criar sistemas mais complexos que englobam todas estas as funcionalidades. Ao automatizar quer os processos de contacto ao consumidor, quer os processos de negócio interno, o ERP e o CRM trabalham juntos para melhorar a eficiência de processos da empresa. Em conclusão, quando estes sistemas que representam a 'front end' e 'back end' da empresa trabalham em conjunto e comunicam entre si, existe uma melhoria geral de todas as funções do negócio.

O crescimento das redes sociais e as suas capacidades de 'networking' criaram um novo desafio envolvendo questões operacionais e de gestão para as empresas superarem. Adotar as capacidades sociais para um ERP não é apenas adicionar um ícone do Facebook ou outra rede social de forma a facilitar os empregados a manter as seus redes de contactos enquanto estão a processar uma transação. A integração do ERP nas redes sociais tem de ser mais sistemática e compreensiva. Segundo Weston (2008) se realizada corretamente esta integração com as redes sociais, pode ajudar a encontrar talento, criar notoriedade de marca, encontrar novos consumidores e ajudar a realizar pesquisas de mercado.

Um exemplo disto é uma aplicação de CRM para ERP que como descrito nos pontos anteriores é um ponto chave num negócio. Greco e White (2009) desenvolveram um conceito denominado por Vendas 2.1, que é uma combinação de CRM, ERP, Web 2.0 e Vendas 2.0 para melhorar a experiência de compra dos clientes. A ideia por de trás do 'Vendas 2.1' é a melhor conexão entre os vendedores e os clientes através de tecnologias



de informação e ERP. Ao adotar sistemas de ERP e aplicações CRM, as empresas podem gravar expectativas, ideias e comentários que os seus clientes deixam nas redes sociais permitindo conseqüentemente que estes dados sejam analisados. Esta conexão com as redes sociais pode ainda beneficiar certos módulos do ERP como os recursos humanos, podendo auxiliar na contratação. (Quast, L. 2012).

No entanto esta integração entre ERP e redes sociais é mais difícil de criar na prática do que na teoria. A sua criação tem que passar por dois desafios: a natureza ágil das redes sociais e a complexidades de dados e processos dos ERP. A grande funcionalidade do ERP é fornecer dados de alta qualidade para todos os sistemas e departamentos da empresa. No entanto os dados extraídos das redes sociais podem não chegar a esses 'standards' e ser difíceis de manipular. Por outro lado, a arquitetura e os processos predefinidos dos ERPs, tem como conseqüência a menor adoção de capacidades sociais. Como resultado e apesar da integração em teoria ser simples, ainda existem vários obstáculos a ser resolvidos. (Greco & White 2009)

Em conclusão, é certo que o futuro dos ERP passa cada vez mais por uma integração com capacidades sociais que respondam às necessidades do negócio. No entanto a complexidade destes sistemas tonará este processo lento e difícil de atingir.

## **2.7 Síntese**

Os sistemas de informação podem ser definidos como um conjunto de componentes inter-relacionados que colecionam ou recuperam, processam, armazenam e distribuem informação para suportar o processo de decisão numa organização. A integração destes sistemas é por norma complexa e depende de uma integração vertical, mas também tem que consistir numa integração horizontal de forma a suportar os processos de negócio. Estes sistemas podem ser divididos em 10 tipos e cada um deles deve seguir uma framework específica para uma integração bem sucedida. Um dos tipos de sistemas de informação é o ERP ou Enterprise Resource Planning.

O Enterprise resource planning é um sistema de informação que possibilita um agrupamento e a interligação de vários módulos numa organização empresarial tendo em comum a sua base de dados. Um dos ERPs mais conhecidos é o SAP. O SAP tem sede em Walldorf na Alemanha, e é a maior fornecedora de aplicações corporativas e

umas das maiores empresas de software a nível mundial. O SAP não só combina as funções do negócio, como permite, ainda, a criação de addons que correm dentro da aplicação ‘mãe’ podendo ter como objetivo a criação de funcionalidades nos processos de negócio até à ligação dos mesmos às redes sociais.

Redes sociais podem ser definidas como serviços web que permitem que pessoas (utilizadores), construam perfis que podem ser públicos ou semipúblicos dependendo das restrições impostas, permitindo ainda, definir a lista de pessoas com que pretendem interagir, ver e cruzar todas as conexões criadas por outros dentro desse sistema. O Facebook e outras redes sociais oferecem uma percentagem de retorno alta, pois permite chegar a muito publico sem utilizar grandes recursos financeiros. Uma das formas que as redes sociais podem ser utilizadas pelas empresas é através de CRM.

O CRM pode ser definido como uma estratégia central de negócio que integra processos internos, funções e redes externas, para criar valor em consumidores chave, retornando lucro. Existem três tipos de CRM: Estratégico, Operacional, e Analítico.

A relação entre ERP, e CRM consiste em serem ambos sistemas desenhados para automatizar e melhorar os processos de negócio. Enquanto o CRM é um sistema para gerir a imagem do negócio, o ERP gere os sistemas na retaguarda do negócio. A relação entre estes sistemas é necessária para trabalharem em conjunto um com o outro de modo que ambos funcionem de forma eficiente e eficazmente. A integração destes sistemas com redes sociais não é apenas adicionar um ícone do Facebook ou outra rede social. A integração tem de ser mais sistemática e compreensiva e, se realizada corretamente, pode ajudar a encontrar talento, criar notoriedade de marca, encontrar novos consumidores e ajudar a realizar pesquisas de mercado.

O futuro dos ERP passa cada vez mais por uma integração com capacidades sociais que respondam às necessidades do negócio. No entanto a complexidade destes sistemas fará este processo lento e difícil de atingir.

As Figuras 10 e 11 demonstram os temas abordados neste capítulo e os autores respectivamente.

Revisão da Literatura	Laudon & Laudon 2006	Granta, Hwanga, Tuc 2013	Stair & Reynolds 2013	Hasselbrin g 2000	Leon 2008	Costa, Resende, Nakamura &	Bolotaeva & Cata 2010	Boyd & Ellison 2007	Barr 2013	Lorette 2014
Sistemas de Informação	Definição		Explicação do funcionamento							
Integração de Sistemas				Como integrar estes sistemas		Integração mediante SI específicos				
ERP		Fases de Integração de ERP			Definição e explicação					
Redes Sociais							Vantagens Financeiras	Definição e explicação		
CRM										
Relação entre ERP, CRM e									ERP nesta relação	Relação entre os sistemas

Figura 10: Revisão dos temas da literatura (parte 1)

Revisão da Literatura	Buttle 2008	Rainer & Cegielski 2010	Chari & Seshadri 2004	Lepoint 2012	Kumar 2010	Sabooniha, Toohey & Lee 2012	Pinto 2010	Schwaiger & Abmayer 2013	Weston 2008	Greco & White 2009
Sistemas de Informação		Tipos de sistemas								
Integração de Sistemas			Framework de Integração			Abordagens na integração		Framework		
ERP							Definição			
Redes Sociais				Marketing no Facebook						
CRM	Definição/CRM Operacional				CRM Estratégico					
Relação entre ERP, CRM e									Vantagens desta relação	Exemplos

Figura 11: Revisão dos temas da literatura (parte 2)

### 3. Modelo Conceptual

#### 3.1 A Arquitetura do Sistema

Neste capítulo é realizado um dos objetivos deste projeto: a criação de uma proposta de um modelo conceptual que permita a integração de ERP com redes sociais.

Para responder a este problema e concretizar o objetivo proposto, foi considerada a arquitetura de integração de sistemas desenvolvida por Chari & Seshadri (2004). O modelo representado na Figura 12 demonstra a relação entre os diferentes sistemas aplicando essa mesma arquitetura.

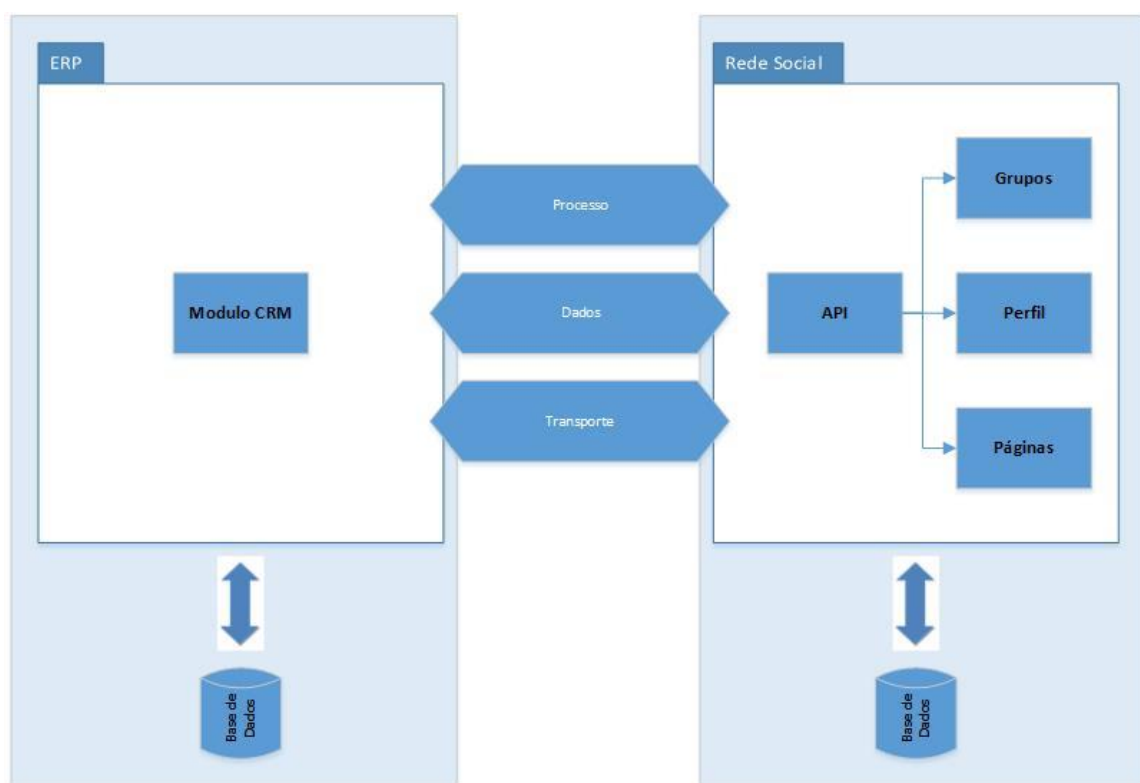


Figura 12: Demonstração do sistema

O modelo da Fig. 12 é composto por dois sistemas. O primeiro sistema é o ERP que é um sistema composto por vários módulos (Pinto 2010), no entanto o módulo de CRM é o essencial para a criação desta aplicação. Por outro lado, as redes sociais são constituídas por perfis, que podem ser públicos ou semipúblicos, e os seus componentes podem ainda variar consoante a rede social em questão, podendo ainda ter grupos, páginas, entre outros (Boyd & Ellison, 2007).

Como o modelo criado é baseado na framework de Chari & Seshadri (2004) este possui três dimensões.

A primeira dimensão denominada de Arquitetura Aplicacional é composta por três componentes. A primeira é a lógica da apresentação. Aqui a aplicação a ser desenvolvida deverá apenas ser acessível em forma de um addon dentro do próprio ERP. A segunda ou lógica de negócio irá ser desenvolvida mais a frente recorrendo a utilização de UML, pois um dos objetivos desta linguagem é dar suporte e tentar simplificar todo o processo de desenvolvimento (Rumbaugh.; Jacobson & Booch, 2005). A última componente desta dimensão é a lógica de dados. Aqui e como se trata de um ERP ou seja um conjunto de vários módulos interligados tendo em comum a sua base de dados (Pinto 2010), a aplicação deverá seguir os standards da base de dados no ERP em questão.

A segunda dimensão é denominada por Nível de Integração e tal como a primeira é composta por três componentes. A primeira é ao nível do processo, isto é, a aplicação integrada no sistema de ERP deverá ser capaz de comunicar diretamente com a Rede Social selecionada, desde de que siga as regras e lógica do negócio. A segunda é o Formato de Dados. Aqui o formato de dados a ser utilizado para comunicar entre sistemas dependerá das especificações da API da rede social ou seja: tipo, estrutura, quantidade de dados entre outros. A componente final da segunda dimensão denominada por Transporte deverá ser responsável pelo transporte de dados entre os sistemas, logo, e como um destes sistemas se trata de uma rede social, deverão existir protocolos http que permitam o envio e receção de dados.

A terceira dimensão refere-se à Especificidade do Domínio da Indústria. Aqui os standards deverão ser genéricos como SQL e http para permitirem a exploração de outros sistemas.

O restante detalhe do sistema irá agora ser descrito como referido previamente em UML, para melhor compreensão a nível de desenvolvimento.

### 3.2 Funcionalidades do sistema

Os esforços das empresas em conduzir integração de sistemas, a reconfigurar processos de negócio e a implementar enterprise resource planning (ERP) permitiram criar uma infraestrutura tecnológica sólida com bases de dados de informação com muita potencialidade e os mais recentes avanços tecnológicos viabilizam o CRM como uma tecnologia implementável e no ponto de vista financeiro bem vinda (Fayerman 2002). Assim partindo deste pressuposto criou-se um conjunto de funcionalidades tendo em conta a natureza de ambos os sistemas (ERP e Redes Sociais). Estas funcionalidades são demonstradas através de um Use Case.

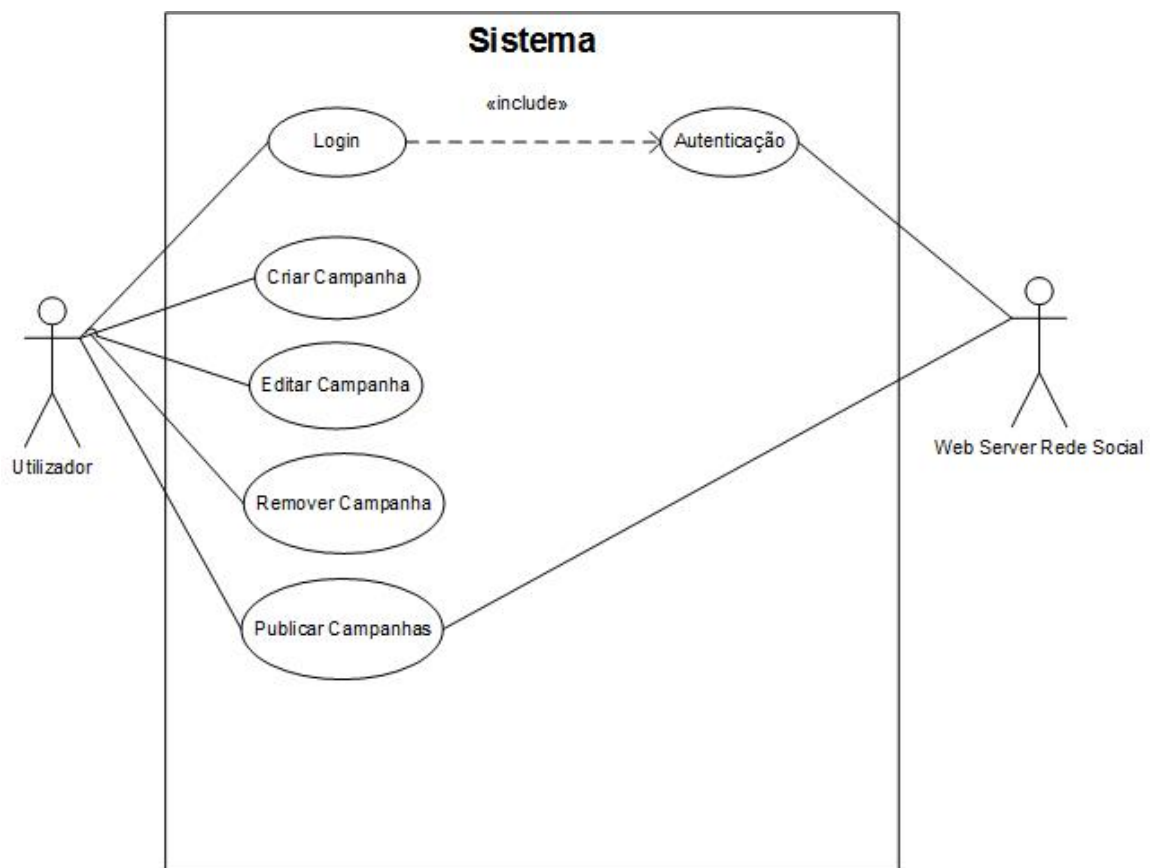


Figura 13: Use Case do Sistema

Como é possível verificar na Figura 13 o modelo tem cinco funcionalidades: Login - ou seja a aplicação deverá obrigatoriamente permitir o Login na rede social; Criar Campanha - permitindo assim a criação de campanhas de marketing utilizando os dados de ERP; Editar Campanha - para permitir a edição de dados de campanhas já criadas; Remover Campanha - para ser possível ao utilizador a remoção das campanhas; Publicar Campanha - funcionalidade que deve permitir ao utilizador publicar as

campanhas na rede social. Cada funcionalidade representada no Use Case é seguidamente detalhada num diagrama de classes e em diagramas de atividades.

### 3.3 Modelo de Classes

O modelo de classes permite apresentar uma visão geral da solução proposta, detalhando assim de modo geral o sistema pretendido.

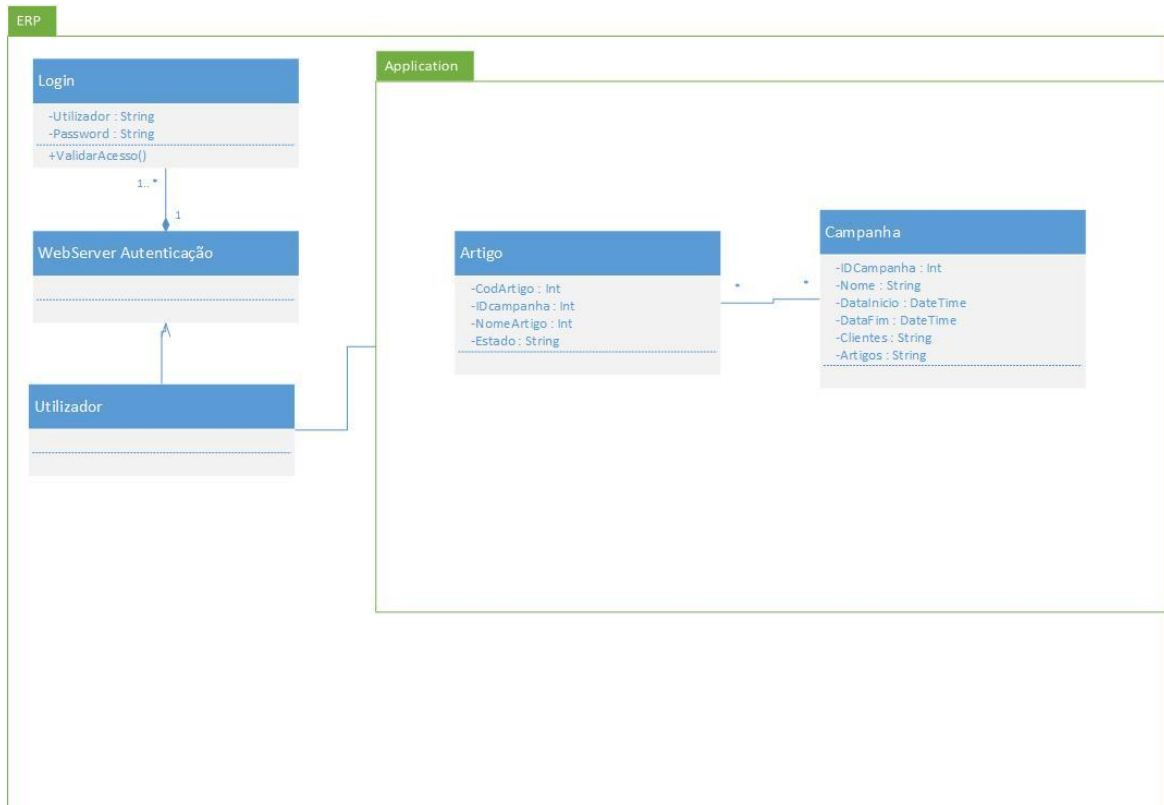


Figura 14: Diagrama de Classes

A aplicação deverá ser iniciada com uma página de login em que o utilizador se autenticará. Após esse processo ser concluído com sucesso, o utilizador deverá ter então a sua disposição todas as funcionalidades previstas no Use Case (Fig. 13). O diagrama de classes acima representado (Figura 14) demonstra como deverá ser estruturada a aplicação.

A primeira classe é denominada por Login. Esta é a classe responsável pelo processo de autenticação. No entanto, este processo é relativo à rede social que venha a ser selecionada, pois o utilizador só deverá poder aceder às funcionalidades do programa, quando tiver todas as autorizações e componentes necessárias por parte da rede social para efetuar publicações na mesma. A classe Login está diretamente ligada ao

WebServer de autenticação. Esta relação está de muitos para um significando que poderá existir mais do que um login para entrar na rede social.

Após o utilizador ser autorizado a aceder às funcionalidades do programa, este deve estar estruturado em duas classes: Artigos e Campanha.

A classe Artigos deverá ser responsável por conter os artigos pertencentes às campanhas que serão criadas. Esta classe deverá possuir os requisitos mínimos necessários como o ID do artigo, nome, ID da campanha a que pertence e o estado do artigo i.e. se está ativo ou não. Esta classe está diretamente relacionada com a classe Campanha numa relação de muitos para muitos ou seja, pode existir várias campanhas para cada artigo e cada artigo pode ter múltiplas campanhas.

A classe campanha deverá ser constituída por IDCampanha, Nome, Data de Inicio, Data de Fim, Clientes, Artigos e deverá ser possível adicionar, modificar ou remover campanhas.

Este diagrama de classes representa um modelo geral de como deverá ser a estrutura da aplicação. Poderá ainda ser necessário adicionar classes específicas de forma a adaptar este modelo a diferentes cenários dependendo, nomeadamente, da especificidade do Enterprise Resource Planning selecionado, da sua estrutura de dados e da rede social escolhida.



### 3.4 Descrição das Funcionalidades

De forma a descrever o detalhe das funcionalidades recorreu-se aos diagramas de atividades para cada funcionalidade presente no Use Case.



Figura 15: Diagrama de Atividades Login

A funcionalidade de Login, se realizada com sucesso deverá, como explicado previamente, dar acesso à aplicação aos seus utilizadores. O utilizador inicia a aplicação e deverá ser confrontado com um sistema de Login da rede social selecionada. Após o preenchimento dos dados o sistema deverá verificar a veracidade dos dados.

Se a resposta do Webserver for positiva, i.e. se o login estiver correto, então o utilizador deverá ser redirecionado para o menu principal da aplicação. Se por outro lado, a resposta do servidor for negativa, deverá ser apresentado ao utilizador a mesma página de login para reintrodução dos dados.



Figura 16: Diagrama de Atividades Criar Campanha

A função Criar Campanha deverá ser selecionada pelo o utilizador a partir do menu principal da aplicação. Após esta função ser selecionada, o utilizador deverá ser redirecionado para uma nova página criada especialmente para esta função. Esta nova página deverá possuir um formulário com todos componentes necessários para a criação de uma nova campanha, sendo que será aqui que o utilizador irá preencher todos os dados como nome da campanha, descrição e datas. Após o preenchimento dos dados da campanha, o utilizador deverá poder adiciona-la. Quando esta funcionalidade for ativada, o sistema deverá gravar na base de dados a nova campanha criada, mas antes, deverá ter que realizar uma validação a todos os dados introduzidos, não só para garantir que o utilizador preencheu todos os requisitos necessários, como também, para garantir que os dados estão em conformidade com as características da tabela na base de dados. Se está validação for realizada com sucesso, uma função deverá adicionar à base

de dados o registo, se por outro lado, algum dos dados for inválido, o utilizador deverá preencher corretamente o formulário e reenviar de novo para aprovação.

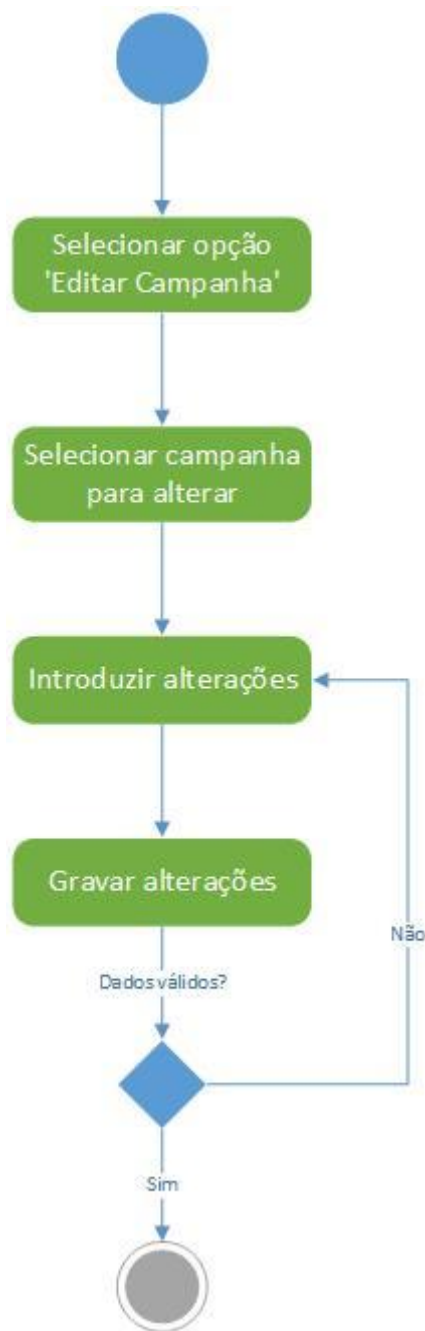


Figura 17: Diagrama de Atividades Editar Campanha

A funcionalidade Editar Campanha deverá ser selecionada no menu principal da aplicação. O utilizador deverá ser redirecionado para uma página criada especificamente para a edição de campanhas. Aqui, deverá ser possível visualizar todas as campanhas presentes na base de dados e onde o utilizador poderá selecionar a campanha que pretende modificar.

Após ter selecionado a campanha a modificar, o utilizador deverá ser redirecionado para uma página que permita visualizar todos os dados passíveis de alteração. Quando as alterações pretendidas forem efetuadas, estas deverão ser gravadas. Depois de se chamar a função para gravar, deverá ser realizada uma verificação de dados para garantir a conformidade com os requisitos necessários, tal como no processo de criação de uma nova campanha. Se o processo de validação for positivo, deverá ser realizado um update na base de dados. Se negativo, o utilizador deverá ter que corrigir os dados que não se encontrem corretos.



Figura 18: Diagrama de Atividades Remover Campanha

A funcionalidade de Remover Campanha deverá ser selecionada no menu principal da aplicação. Deverá ser apresentado ao utilizador uma listagem com todas as campanhas presentes na base de dados, onde o utilizador poderá selecionar a campanha pretendida para remoção.

Quando a campanha pretendida for selecionada o utilizador apenas terá que ativar a função de remover a campanha através da interface gráfica para que esta seja removida da base de dados.



Figura 19: Diagrama de Atividade Publicar Campanha

A funcionalidade Publicar Campanha deverá ser selecionada no menu principal da aplicação. O utilizador deverá ser redirecionado para uma página criada especificamente para esta publicação. Aqui o utilizador poderá selecionar a campanha pretendida para ser publicada.

Após ter sido selecionada, a campanha deverá ser então enviada para a rede social escolhida, mas terá que passar por uma validação de dados para apurar, se por exemplo, não se está a publicar uma campanha cuja data final já tenha expirado.

Se o resultado desta verificação for positiva, a campanha poderá seguir para publicação na rede social. Na eventualidade de esta validação não ser positiva, o utilizador terá que alterar os dados necessários para a publicação ser autorizada.

## **4. Desenvolvimento do Protótipo**

### **4.1 Introdução**

Com vista a verificar a exequibilidade da solução proposta foi desenvolvido um protótipo. Para o desenvolvimento deste protótipo foi selecionado o ERP SAP Business One. O SAP Business One permite a criação de addons, ou seja, a criação de aplicações que vão correr dentro do próprio ERP. Estas aplicações podem ser criadas através de duas linguagens de programação suportadas pelo SAP B1: Visual Basic (VB) e C Sharp (C#). Para a criação e desenvolvimento de código do protótipo foram utilizados o Visual Studio 2013 e a linguagem C#.

Na criação de um addon, ou seja uma aplicação que irá correr dentro de outra, é necessário ter em conta os aspetos visuais. Sendo possível utilizar o Visual Studio para a criação dos elementos visuais da aplicação como caixas de texto ou botões, estes não estariam em conformidade com o restante aspeto visual do SAP. Para atingir esse fim foi utilizado uma ferramenta disponibilizada pelo o próprio ERP denominada de *'ScreenPainter'*. Através desta ferramenta é possível criar formulários com botões, caixas de texto e outros objetos em conformidade visual com o resto da aplicação.

Por fim a rede social selecionada foi o Facebook. De forma a interagir com esta rede, foi necessária a implementação da sua API no programa, de forma a este poder comunicar com sucesso com o servidor do Facebook.

### **4.2 Requisitos**

A aplicação a ser desenvolvida tem os seguintes requisitos funcionais: criação, edição, remoção e publicação de campanhas publicitárias.

A um nível mais detalhado a aplicação deverá seguir uma estrutura predefinida. Deverá começar por apresentar um menu de login no Facebook de modo ao utilizador se autenticar. Após esta autenticação ser realizada com sucesso, o utilizador deverá poder seleccionar que ação pretende realizar dentro do programa. Estas ações são os requisitos funcionais referidos no parágrafo anterior.

A criação de campanhas terá de ser totalmente especificada i.e. o utilizador deverá ter que preencher os seguintes campos: nome, descrição, data de início, data de fim, grupos de clientes, grupos de artigos e artigos a não publicar. Estes últimos três campos

prendem-se, diretamente, com a estrutura que o SAP utiliza para organizar os dados dentro do ERP.

Os parceiros de negócio dentro do ERP, são automaticamente agrupados por classes (ex. Clientes, Fornecedores, Grandes Contas). Cada uma destas classes ou grupos contem todos os parceiros de negócios específicos (ex. Grupo de Fornecedores: Fornecedor A, Fornecedor B ...). Os artigos seguem a mesma lógica, e, também estão divididos por grupos. (ex. Impressoras, Computadores, Servidores).

O utilizador deverá poder selecionar múltiplos grupos de artigos e clientes aquando da criação da campanha. O ultimo campo 'artigos a não publicar' prende-se com o utilizador poder selecionar que artigos não deseja publicar dentro dos próprios grupos i.e. se o utilizador selecionou o grupo 'Impressoras', deverá ter um mecanismo que permita a possibilidade de selecionar quais as impressoras que fazem e não fazem parte da campanha.

A ação de edição de campanha deverá ser desenvolvida de forma a permitir a edição de todos os seus dados, enquanto a ação de remover deverá eliminar qualquer registo relativo à campanha selecionada para remoção.

Todas as campanhas deverão ser publicadas em 'grupos de Facebook' mediante os grupos de clientes selecionado i.e. para cada grupo de clientes do ERP deverá ser associado a um grupo dentro da rede social. Desta forma se o utilizador tiver selecionado 4 grupos de clientes, a campanha deverá ser publicada em 4 grupos no Facebook respetivamente.

### **4.3 Desenvolvimento**

Antes de qualquer criação de uma interface gráfica foi necessário criar a estrutura de dados dentro da base de dados SAP de forma a poder armazenar todas as campanhas e dados complementares.

Para este fim, foram criadas 4 tabelas na base de dados (Tabela 3, 4, 5, 6): tabela das campanhas, dos clientes, dos artigos e dos artigos a não publicar. A primeira tabela foi criada para armazenar os dados da campanha, enquanto as restantes foram criadas para armazenar os grupos que o SAP cria, matéria já anteriormente desenvolvida.



**Tabela 3: Tabela das Campanhas.**

<b>Campo</b>	<b>Descrição</b>	<b>Tipo</b>
Numero	Nº da campanha	Numérico(10)
Nome	Nome da campanha	Alfanumérico(20)
Descrição	Descrição da campanha	Alfanumérico(200)
Datainicio	Data de inicio da campanha	Data
Datafim	Data de fim da campanha	Data
Gclientes	Nº de grupos de clientes	Alfanumérico(40)
Gartigos	Nº de grupos de artigos	Alfanumérico(40)
Publicado	Indica se já foi publicado ou não	Alfanumérico(10)

**Tabela 4: Tabela dos Grupos Clientes.**

<b>Campo</b>	<b>Descrição</b>	<b>Tipo</b>
Numero	Nº do registo na tabela	Numérico(10)
Grupo	Nº do grupo de clientes	Alfanumérico(10)
Campanha	Nº da campanha	Alfanumérico(10)

**Tabela 5: Tabela dos Grupos Artigos.**

<b>Campo</b>	<b>Descrição</b>	<b>Tipo</b>
Numero	Nº do registo na tabela	Numérico(10)
Grupo	Nº do grupo de artigos	Alfanumérico(10)
Campanha	Nº da campanha	Alfanumérico(10)

**Tabela 6: Tabela dos Artigos a não publicar.**

<b>Campo</b>	<b>Descrição</b>	<b>Nome</b>
Numero	Nº do registo na tabela	Numerico(10)

Artigo	Código do artigo	Alfanumerico(10)
Nome	Nome do artigo	Alfanumerico(50)
Preço	Preço do artigo	Alfanumerico(10)
Campanha	Nº da campanha	Alfanumerico(10)
Gartigo	Nº do Grupo de Artg.	Alfanumérico(10)
Estado	Define se o artigo é publicado ou não	Alfanumérico(10)

De forma a aplicação poder interagir com o Facebook é necessário criar uma *app* dentro da Rede Social. Esta *app* é uma ferramenta que irá obter as permissões necessárias e será responsável pelas publicações. Desta forma a aplicação que irá ser criada no SAP apenas terá que enviar, utilizando a API do Facebook, todos os dados relativos para realizar as publicações (mensagem, grupos onde publicar).

A aplicação para o SAP começou a ser desenvolvida pelo sistema de autenticação. Este sistema consiste numa página Web que é carregada quando o utilizador inicia a aplicação Figura 20.



Figura 20: Sistema de Autenticação

Esta página de autenticação tem de ser específica, pois tem de efetuar o login na *app* criada previamente. Para isso o seguinte código foi criado.

```
20
21
22
23
24
25
26
27
28
29
30
31
private void button1_Click(object sender, EventArgs e)
{
    pictureBox1.Visible = false;
    button1.Visible = false;
    button2.Visible = false;
    webBrowser1.Navigate("https://graph.facebook.com/oauth/authorize" +
        "?client_id=1388642484737115" +
        "&redirect_uri=http://www.facebook.com/connect/login_success.html" +
        "&display=popup&type=user_agent&scope=user_status,user_photos,email," +
        "user_birthday,user_online_presence,publish_stream");
}
```

Figura 21: Função responsável pelo load da página de autenticação

O código da Figura 21 é responsável por pedir as credenciais específicas para fazer login na *app* criada. Isto é, realizado através de um link específico, onde está presente o ID da *app*.

Quando a autenticação for realizada com sucesso um novo menu é iniciado permitindo ao utilizador a escolha da ação que pretende efetuar na aplicação Figura 22. Aqui o utilizador pode seleccionar três opções, cada uma abrindo uma nova página na aplicação. Neste menu o utilizador pode realizar as 4 ações previstas nos requisitos, sendo que a opção de criar, editar e remover se encontram no mesmo formulário.

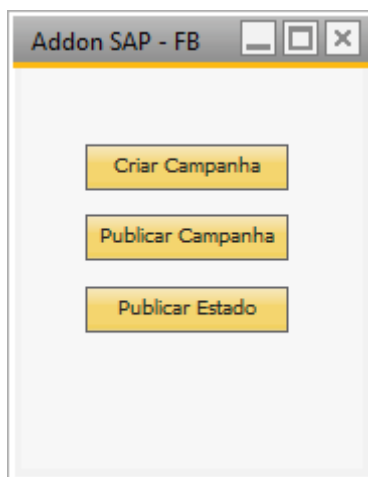


Figura 22: Menu Principal

Para a primeira ação, criação de campanha, foram colocadas várias caixas de texto para o utilizador preencher alguns dos requisitos necessários para a criação de uma campanha i.e. nome, descrição, data de inicio, data de fim. Como para a criação bem sucedida de uma campanha são necessários todos os campos preenchidos e não apenas

os referidos no paragrafo anterior, sendo por isso necessário criar um sistema de escolha múltipla devido aos grupos de clientes e artigos referidos no inicio deste capitulo (Figura 23).

The screenshot shows a software window titled "Criar Campanha". It contains several input fields and controls:

- Nome:** A text input field.
- Descrição:** A larger text input field.
- Data Inicio:** A date input field.
- Data Fim:** A date input field.
- Grupo Clientes:** A dropdown menu with a list of items below it. Below the list are "Adicionar" and "Remover" buttons.
- Grupo Artigos:** A dropdown menu with a list of items below it. Below the list are "Adicionar", "Remover", and "Excepções" buttons.
- Table:** A table with 7 columns: "Nº:", "Nome:", "Descrição:", "Data Inicio:", "Data Fim:", "Grupo Clientes:", and "Grupo Artigos:". The table is currently empty.
- Buttons:** At the bottom right, there are three buttons: "Adicionar" (positioned above the table), "Editar", "Remover", and "Sair".

Figura 23: Formulário de criação de campanha

O uso deste sistema é simples e intuitivo. As caixas de seleção estão configuradas para listar todos os grupos de Clientes e Artigos, consoante o campo que o utilizador está a preencher (Figura 24), permitindo assim adicionar e remover múltiplos grupos de clientes e artigos da campanha.

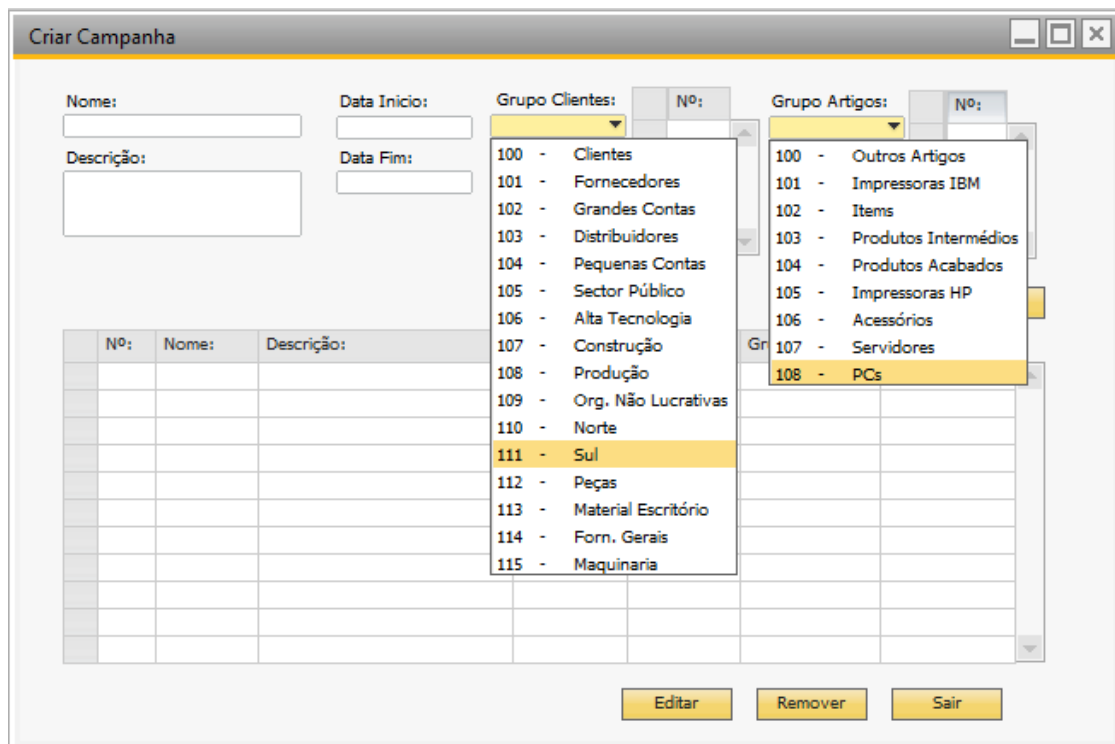


Figura 24: Sistema de escolha múltipla

O preenchimento das caixas é realizado usando o seguinte código:

```

1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
...
private void addGClientes(SAPbouiCOM.ComboBox oCombo)
{
    SAPbobsCOM.Recordset RS;

    RS = oCompany.GetBusinessObject(SAPbobsCOM.BoObjectTypes.BoRecordset);

    RS.DoQuery("SELECT GroupCode,GroupName FROM OCRG ORDER BY GroupCode");

    RS.MoveFirst();
    while (RS.EOF == false)
    {
        oCombo.ValidValues.Add(RS.Fields.Item("GroupCode").Value, RS.Fields.Item("GroupName").Value);
        RS.MoveNext();
    }
}

```

Figura 25: Preenchimento das caixas de seleção

O código acima executa um query específico consoante a caixa de seleção. No caso da caixa de grupo de clientes o query executado é "SELECT GroupCode, GroupName FROM OCRG ORDER BY GroupCode". Este query permite ir buscar à base de dados todos os códigos de grupo e o nome desses grupos organizando-os, obtendo o resultado visível da Figura 24.

Por fim foi criado um mecanismo que permite ao utilizador seleccionar os artigos que irão ou não irão ser publicados. Se o utilizador adicionar, por exemplo, o grupo de artigos '101 - Impressoras', todos os artigos pertencentes a esse grupo estarão

disponíveis num outro formulário. O utilizador através desse formulário poderá seleccionar os artigos a publicar (Fig. 26).

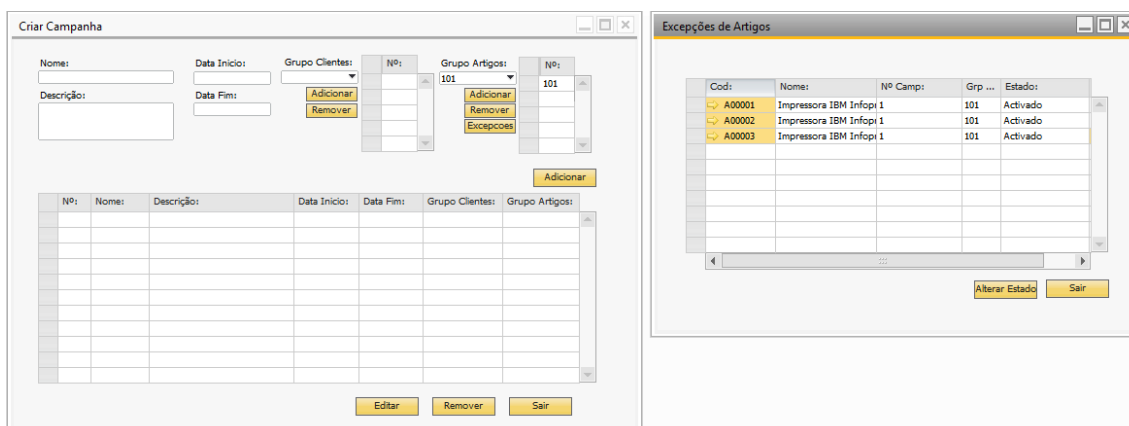


Figura 26: Sistemas de exceções de artigos

Quando o utilizador termina de preencher todos os campos e adiciona a campanha, essa mesma campanha é listada numa matriz já existente.

A adição da campanha é realizada através do código na Figura 27.

```

1242     UTS = oCompany.UserTables;
1243     UT = UTS.Item("TESE");
1244     UT.Code = System.Convert.ToString(i);
1245     UT.Name = oEdit.String;
1246     UT.UserFields.Fields.Item("U_Numero").Value = i;
1247     UT.UserFields.Fields.Item("U_Nome").Value = oEdit.String;
1248     UT.UserFields.Fields.Item("U_Descricao").Value = ExtEdit.String;
1249     UT.UserFields.Fields.Item("U_Datainicio").Value = oEdit1.String;
1250     UT.UserFields.Fields.Item("U_Datafim").Value = oEdit2.String;
1251     UT.UserFields.Fields.Item("U_Gclientes").Value = campoclientes;
1252     UT.UserFields.Fields.Item("U_Gartigos").Value = campoartigos;
1253     UT.Add();

```

Figura 27: Adição da campanha

Este código não representa a totalidade da função que adiciona a campanha pois é extensa. No entanto é onde todos os campos que o utilizador introduziu são adicionados a tabela.

Na segunda ação, edição de campanha, o utilizador selecciona uma campanha já criada para editar, todos os dados da campanha seleccionada são carregados para o formulário, onde o utilizador os pode alterar. Para gravar as alterações o utilizador simplesmente tem de pressionar o botão ‘Gravar’ (Figura 28).

Nome: Natal Feliz  
 Descrição: Este natal aproveite ...  
 Data Inicio: 20131223  
 Data Fim: 20131225  
 Grupo Clientes: 102  
 Grupo Artigos: 105

Nº:	Nome:	Descrição:	Data Inicio:	Data Fim:	Grupo Clientes:	Grupo Artigos:
1	Natal Feliz	Este natal aproveite ...	23/12/2013	25/12/2013	100 102	101 105

Figura 28: Edição de uma campanha

A terceira ação, remover campanhas, acontece quando uma linha da matriz é selecionada e o botão 'Remover' é pressionado. Quando isto acontece a campanha selecionada é removida tal como todos os dados associados à mesma em todas as tabelas. Isto é executado através do seguinte código.

```

1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
}

if (oRecordset.RecordCount > 0)
{
  oRecordset.MoveFirst();

  sSQL = System.Convert.ToString(oRecordset.Fields.Item("U_Numero").Value);

  UTS = oCompany.UserTables;
  UT = UTS.Item("TESE");
  UT.GetByKey(sSQL);
  UT.Remove();
  SBO_Application.MessageBox("Campanha Removida", 1, "OK", "", "");
}

//RemoverGartigo(a);
RemoverGclienteTotal(a);
RemoverGartigoTotal(a);
RemoverTabelaExcepcoesTotal(a);
oMatrix.DeleteRow(j);
}

```

Figura 29: Código edição de uma campanha

A Fig. 29 mostra o código responsável pela remoção do registo da tabela campanhas, e como é possível verificar no final dessa remoção são invocadas três funções, uma para cada tabela i.e. após a remoção dos dados da tabela de campanhas, é chamada uma

função para limpar tudo relacionado com essa campanha na tabela de grupos de clientes outra função irá remover todos os registros da tabela de grupos de artigos e, por fim, a última função eliminará todos os registros da tabela dos artigos a não publicar. Quando a campanha é removida, uma mensagem de sucesso é enviada ao utilizador.

A quarta ação, Publicação de campanhas, foi criada de forma ao utilizador poder facilmente publicar as campanhas que se encontram no sistema. Esta página está incorporada com um sistema de procura, para auxiliar o utilizador no caso de existirem muitos registros na base de dados. Assim o utilizador pode pesquisar uma campanha por nome ou número. A Fig. 30 mostra o algoritmo utilizado para executar a pesquisa, consoante o tipo de pesquisa que o utilizador selecionar i.e. por nome ou por número.

```
if (tipo == "Nome")
{
    for (int i = 1; i <= oMatrix.RowCount; i++)
    {
        if (((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_5").Cells.Item(i).Specific).Value.ToString() == pesquisa)
        {
            oMatrix.SelectRow(i, true, true);
        }
    }
}

if (tipo == "Numero")
{
    for (int i = 1; i <= oMatrix.RowCount; i++)
    {
        if (((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_6").Cells.Item(i).Specific).Value.ToString() == pesquisa)
        {
            oMatrix.SelectRow(i, true, true);
        }
    }
}
```

Figura 30: Algoritmo de pesquisa

O algoritmo de pesquisa automaticamente mostra os resultados obtidos como demonstrado na Figura 31.



Nº:	Nome:	Descrição:	Data Inicio:	Data Fim:	Grupo Clientes:	Grupo Artigos:
1	Natal Feliz	Descrição do	23/12/2013	25/12/2013	100 102	101

Figura 31: Formulário publicar

Para publicar o utilizador seleciona a linha da matriz que pretende e pressiona o botão 'Publicar'. Aqui, antes da publicação, são efetuadas algumas verificações chave. Uma campanha nunca pode ser publicada mais do que uma vez, isto é, se o utilizador tentar publicar a campanha em duplicado, o programa automaticamente bloqueia a publicação. O programa efetua esta mesma ação se o utilizador tentar publicar uma campanha onde a sua data final já tenha expirado i.e. o programa não permite a publicação de campanhas passadas. Quando o utilizador publica a campanha e cumpre todos os requisitos uma mensagem de sucesso é enviada. A Fig. 32 demonstra o código da função publicar.

```

957   oRecordset = ((SAPbobsCOM.Recordset) (oCompany.GetBusinessObject(SAPbobsCOM.BoObjectTypes.BoRecordset))) :
958   string token = "CAATu9oYFIzBaOUqXeB4gOORpk07ppzCBcygSjYxy2AofctNK7s7552CzpguQIawlhtpC88FJ7ooChRyV50zfe28
959   FacebookAPI api = new Facebook.FacebookAPI(token);
960   myToken.Default.userId = grupo.ToString();
961   Dictionary<string, string> postArgs = new Dictionary<string, string>();
962   postArgs["message"] = enviar;
963   JSONObject post = api.Post("/") + grupo.ToString() + "/feed", postArgs);

```

Figura 32: Código responsável por publicar a campanha

Uma ação possível, que não se encontra presente no modelo (Publicar estado), foi desenvolvida de forma ao utilizador poder publicar estados no seu perfil em junção com informações chave do SAP. O utilizador começa por carregar os dados (ex. artigos,

clientes) que pretende usando um menu. Quando este carregamento é efetuado os dados selecionados são listados numa matriz Figura 33.

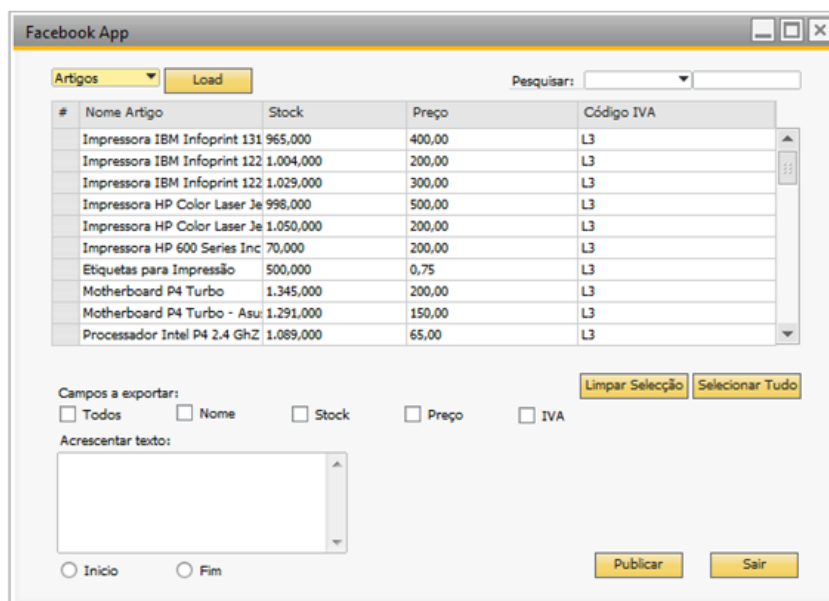


Figura 33: Publicar Estado

Aqui o utilizador pode também utilizar um campo de pesquisa de modo a facilitar a procura por artigos ou clientes específicos. Para publicar um estado o utilizador tem apenas de selecionar a(s) linha(s) e pressionar o botão ‘Publicar. No entanto, toda esta informação pode ser personalizada, isto é, o utilizador embora selecionando uma linha completa, pode escolher através de várias checkboxes, quais os campos exportar dessa mesma linha. Pode ainda acrescentar uma mensagem personalizada para além da informação que selecionou para ser publicada. A Fig. 34 demonstra o código utilizado para publicar as informações no perfil da empresa.

```

2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
try
{
    FacebookAPI api = new Facebook.FacebookAPI(myToken.Default.token);
    JObject me = api.Get("/me");
    var yo = me.Dictionary["id"].String;
    myToken.Default.userId = yo.ToString();
    Dictionary<string, string> postArgs = new Dictionary<string, string>();
    postArgs["message"] = temp2;
    JObject post = api.Post("/" + myToken.Default.userId + "/feed", postArgs);
    oEdit.String = "";
    SBO_Application.MessageBox("Enviada", 1, "OK", "", "");
    temp = "";
    temp2 = "";
    temp3 = "";
}
catch (FacebookAPIException exr)
{
    Console.WriteLine(exr.Message);
}

```

Figura 34: Código Publicar Estado

## 5. Testes funcionais e avaliação preliminar

Foram realizados testes ao protótipo para garantir a sua total funcionalidade. Estes testes foram baseados e executados de acordo com as funcionalidades descritas no Use Case apresentado no Modelo Conceptual (Login, Criar Campanha, Editar Campanha, Remover Campanha, Publicar Campanha).

A avaliação efetuada consiste em descrever as funcionalidades, demonstrar a sua implementação e a sua utilização.

### 5.1 Funcionalidade Login

- **Descrição**

A funcionalidade de Login, se realizada com sucesso, deverá como explicado previamente, dar acesso à aplicação aos seus utilizadores. O utilizador inicia a aplicação e deverá ser confrontado com um sistema de Login da rede social selecionada. Após o preenchimento dos dados o sistema deverá verificar a veracidade dos mesmos. Se estes forem validados, então o utilizador deverá ser redirecionado para o menu principal da aplicação. Se, por outro lado estiverem incorretos, deverá ser apresentado ao utilizador a mesma página de login para reintrodução dos dados.

- **Implementação**

A implementação desta funcionalidade consistiu na criação de uma página inicial desenvolvida em C# em que o utilizador pode selecionar proceder, para a página de login ou sair da aplicação. A página de login, por sua vez consiste num WebBrowser que automaticamente é redirecionado para a página de login da rede social.

- **Utilização**

O utilizador após iniciar a aplicação procede a escolha de prosseguir para a página de login ou sair da aplicação como demonstrado na Figura 35.



Figura 35: Tutorial de página login

O utilizador após seleccionar a opção de Login é apresentado com o formulário para preencher com as suas credenciais como demonstra a Figura 36.



Figura 36: Tutorial introdução dados

Se o login for bem sucedido a página principal da aplicação (Figura 37), deverá ser apresentada.

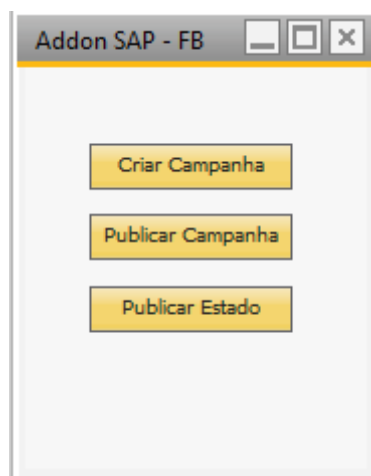


Figura 37: Tutorial menu principal

## 5.2 Funcionalidade Criar Campanha

- **Descrição**

A função Criar Campanha deverá ser selecionada pelo o utilizador a partir do menu principal da aplicação. Após esta função ser selecionada o utilizador deverá ser redirecionado para uma nova página. Esta nova página deverá possuir um formulário, com todos componentes necessários, para a criação de uma nova campanha. Após o preenchimento dos dados da campanha o utilizador deverá poder adicionar a campanha através de um componente de interface gráfica. Quando esta funcionalidade for ativada o sistema deverá gravar na base de dados a nova campanha criada mas, antes, deverá ter que realizar uma validação a todos os dados introduzidos, não só para verificar que o utilizador preencheu todos os requisitos necessários como também para garantir que os dados estão em conformidade com as características da tabela na base de dados. Se esta validação for realizada com sucesso, uma função deverá adicionar à base de dados o registo, se, por outro lado, algum dos dados for inválido, o utilizador deverá preencher corretamente o formulário e reenviar de novo para aprovação.

- **Implementação**

A implementação desta funcionalidade passou por desenvolver uma página

específica para a criação de campanhas. Como as restantes funcionalidades da aplicação, foi desenvolvida em C#. A interface gráfica foi criada no ScreenPainter de forma a respeitar o layout geral do SAP. Esta página foi desenvolvida de forma a que o utilizador preencha todos os dados necessários, estando ao mesmo tempo adaptada para a estrutura de dados específica do SAP.

- **Utilização**

O utilizador deverá seleccionar a opção ‘Criar Campanha’ no menu principal como indicado na Figura 38.

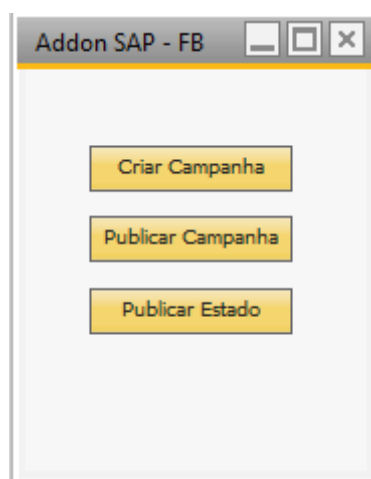


Figura 38: Tutorial menu principal 2

É apresentado de seguida ao utilizador um formulário para preencher. Este formulário é constituído por nome da campanha, descrição, data de inicio, data

de fim, grupo de clientes e grupo de artigos (Figura 39).

The screenshot shows a window titled "Criar Campanha". It contains several input fields: "Nome:" (empty), "Data Inicio:" (empty), "Data Fim:" (empty), "Grupo Clientes:" (dropdown menu), and "Grupo Artigos:" (dropdown menu). Below these are "Adicionar" and "Remover" buttons for each group. A table with 7 columns (Nº, Nome, Descrição, Data Inicio, Data Fim, Grupo Clientes, Grupo Artigos) is visible below the form. At the bottom are "Editar", "Remover", and "Sair" buttons.

Figura 39: Tutorial menu criar campanha

O utilizador deverá preencher todos os campos, sendo que o campo de Grupo Clientes e Grupo Artigos é preenchido usando listas como mostra a Figura 40.

The screenshot shows the "Criar Campanha" form with some fields filled: "Nome:" is "Campanha 1", "Data Inicio:" is "24/08/2015", "Data Fim:" is "24/09/2015", "Grupo Clientes:" is "100", and "Grupo Artigos:" is "100". The "Adicionar" and "Remover" buttons are present. The table below has the same structure as in Figure 39. The "Grupo Artigos:" dropdown menu is open, showing a list of items: "100 - Outros Artigos", "101 - Impressoras IBM", "102 - Items", "103 - Produtos Intermedios", "104 - Produtos Acabados", "105 - Impressoras HP", "106 - Acessórios", "107 - Servidores", and "108 - PCs". At the bottom are "Editar", "Remover", and "Sair" buttons.

Figura 40: Tutorial menu criar campanhas preenchido

O utilizador pode ainda neste menu adicionar ou remover artigos da campanha publicitária (Figura 41).

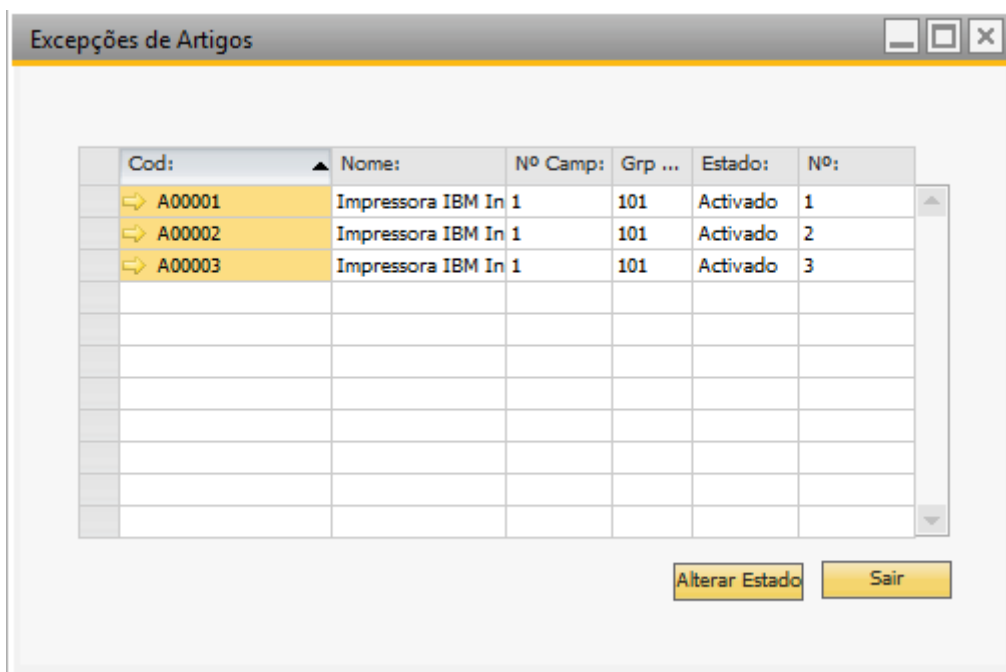


Figura 41: Tutorial adicionar e remover artigos

Após todos os dados estarem devidamente preenchidos e personalizados basta apenas clicar no botão 'Adicionar' para que a campanha seja adicionada à base de dados e poder ser visualizada numa grelha como mostra a Figura 42.

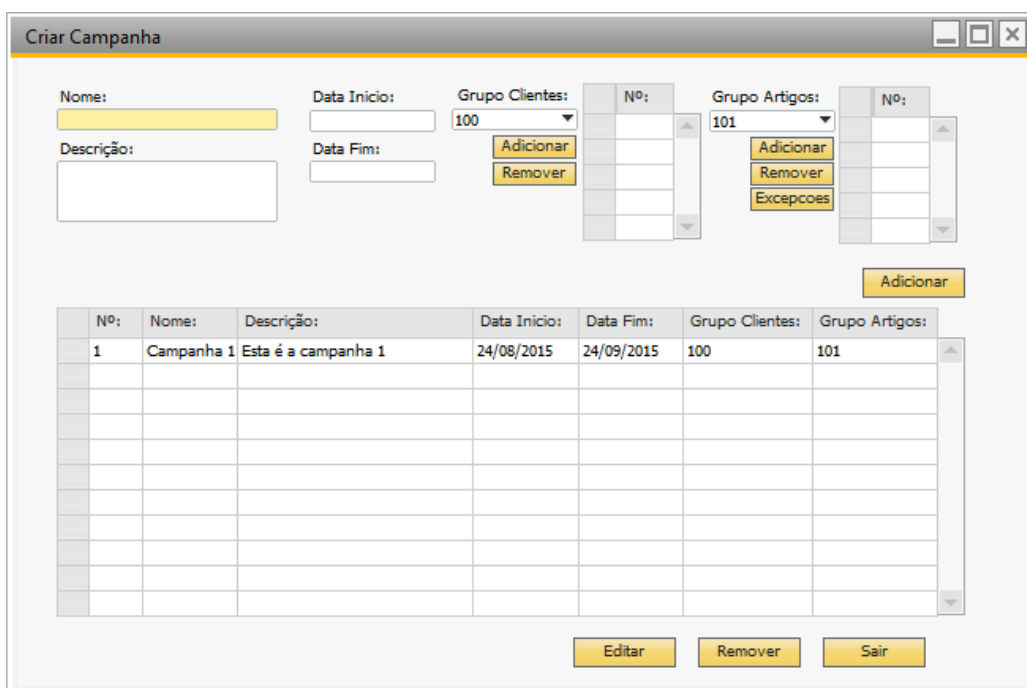


Figura 42: Tutorial campanha adicionada



### 5.3 Funcionalidade Editar Campanha

- **Descrição**

A funcionalidade Editar Campanha deverá ser selecionada no menu principal da aplicação. Aqui deverá ser possível visualizar todas as campanhas presentes na base de dados e será onde o utilizador irá selecionar a campanha que pretende modificar. Após ter selecionado a campanha a modificar, o utilizador deverá ser redirecionado para uma página que permita visualizar todos os dados passíveis de alteração. Quando as alterações pretendidas forem efetuadas, estas deverão ser gravadas.

- **Implementação**

A implementação desta funcionalidade passou por aproveitar a página desenvolvida para a criação de campanhas. Esta funcionalidade foi desenvolvida na mesma página pois são ambas funcionalidades relacionadas com dados de campanha. A implementação desta funcionalidade usando este método, permite ao utilizador uma fácil edição de uma campanha acabada de criar, fazendo as alterações necessárias na mesma página. A interface foi desenvolvida para ser intuitiva e simples tendo o utilizador apenas que selecionar a campanha e editar.

- **Utilização**

O utilizador deverá selecionar a página 'Criar Campanha' pois a função editar encontra-se nesta mesma página.

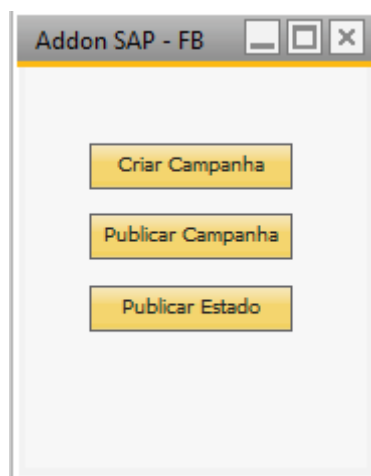


Figura 43: Tutorial menu principal 3

O utilizador seleciona a campanha a editar e pressiona o botão ‘Editar’ como demonstrado na Figura 44.

The screenshot shows the 'Criar Campanha' window with the following fields and controls:

- Nome:** Text input field.
- Descrição:** Text input field.
- Data Inicio:** Date input field.
- Data Fim:** Date input field.
- Grupo Clientes:** Dropdown menu with 'Adicionar' and 'Remover' buttons.
- Grupo Artigos:** Dropdown menu with 'Adicionar', 'Remover', and 'Excepcoes' buttons.
- Table:** A table with columns: Nº, Nome, Descrição, Data Inicio, Data Fim, Grupo Clientes, Grupo Artigos. The first row is highlighted in yellow.
- Buttons:** 'Adicionar' (top right), 'Editar' (bottom center), 'Remover' (bottom right), and 'Sair' (bottom right).

Nº	Nome	Descrição	Data Inicio	Data Fim	Grupo Clientes	Grupo Artigos
1	Campanha 1	Esta é a campanha 1	24/08/2015	24/09/2015	100	101

Figura 44: Tutorial seleção editar campanha

Após este processo o sistema preenche todos os dados passíveis de alteração nos campos devidos, onde o utilizador os poderá alterar (Figura 45).

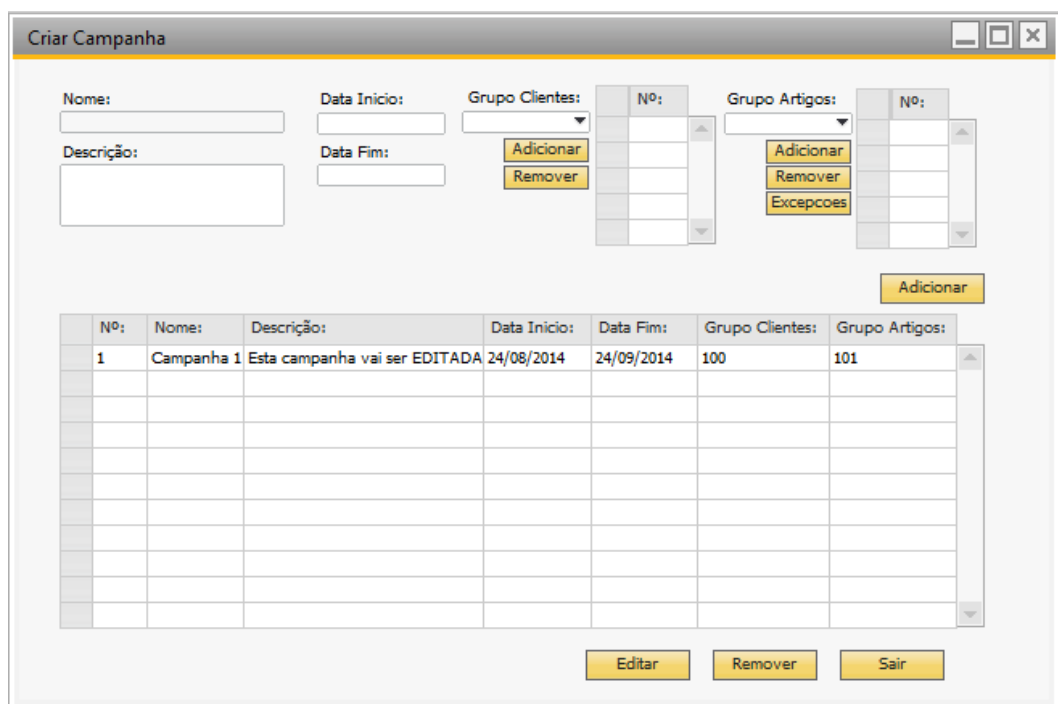
The screenshot shows the 'Criar Campanha' window with the following fields and controls:

- Nome:** Text input field with 'Campanha 1' entered.
- Descrição:** Text input field with 'Esta é a campanha 1' entered.
- Data Inicio:** Date input field with '24/08/2014' entered.
- Data Fim:** Date input field with '24/09/2014' entered.
- Grupo Clientes:** Dropdown menu with 'Adicionar' and 'Remover' buttons.
- Grupo Artigos:** Dropdown menu with 'Adicionar', 'Remover', and 'Excepcoes' buttons.
- Table:** A table with columns: Nº, Nome, Descrição, Data Inicio, Data Fim, Grupo Clientes, Grupo Artigos. The first row is highlighted in yellow.
- Buttons:** 'Adicionar' (top right), 'Gravar' (top right), 'Editar' (bottom center), 'Remover' (bottom right), and 'Sair' (bottom right).

Nº	Nome	Descrição	Data Inicio	Data Fim	Grupo Clientes	Grupo Artigos
1	Campanha 1	Esta é a campanha 1	24/08/2015	24/09/2015	100	101

Figura 45: Tutorial dados editar campanha

Por fim e após as alterações terem sido efetuadas, o utilizador apenas tem que guardar utilizando o botão ‘Gravar’ como demonstra a Figura 45.



The screenshot shows a window titled "Criar Campanha" with the following elements:

- Form Fields:**
  - Nome:
  - Descrição:
  - Data Inicio:
  - Data Fim:
  - Grupo Clientes:  (with dropdown arrow)
  - Grupo Artigos:  (with dropdown arrow)
- Buttons:**
  - Adicionar (next to Grupo Clientes)
  - Remover (next to Grupo Clientes)
  - Adicionar (next to Grupo Artigos)
  - Remover (next to Grupo Artigos)
  - Excepcões (next to Grupo Artigos)
  - Adicionar (bottom right of the main form area)
  - Editar (bottom center)
  - Remover (bottom right)
  - Sair (bottom right)
- Table:**

Nº:	Nome:	Descrição:	Data Inicio:	Data Fim:	Grupo Clientes:	Grupo Artigos:
1	Campanha 1	Esta campanha vai ser EDITADA	24/08/2014	24/09/2014	100	101

Figura 46: Tutorial edição campanha completa

#### 5.4 Funcionalidade Remover Campanha

- **Descrição**

A funcionalidade de Remover Campanha deve ser seleccionada no menu principal da aplicação. Deverá ser apresentado ao utilizador uma listagem com todas as campanhas presentes na base de dados, onde o utilizador poderá seleccionar a campanha pretendida para remoção.

- **Implementação**

A implementação desta funcionalidade passou por aproveitar a página desenvolvida para a criação e edição de campanhas. A implementação desta funcionalidade usando este método, permite ao utilizador uma fácil remoção de qualquer campanha. A interface foi desenvolvida para ser simples, tendo o utilizador apenas que seleccionar a campanha em questão e clicar ‘Remover’.

- **Utilização**

O utilizador deverá seleccionar a página ‘Criar Campanha’ pois a função remover encontra-se nesta mesma página.

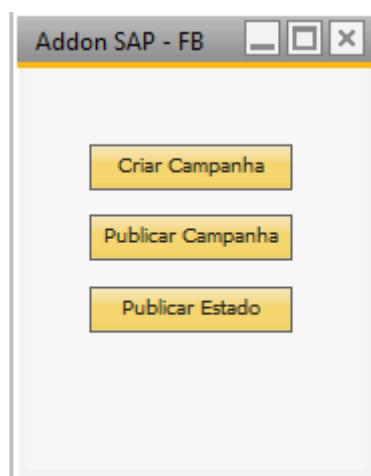
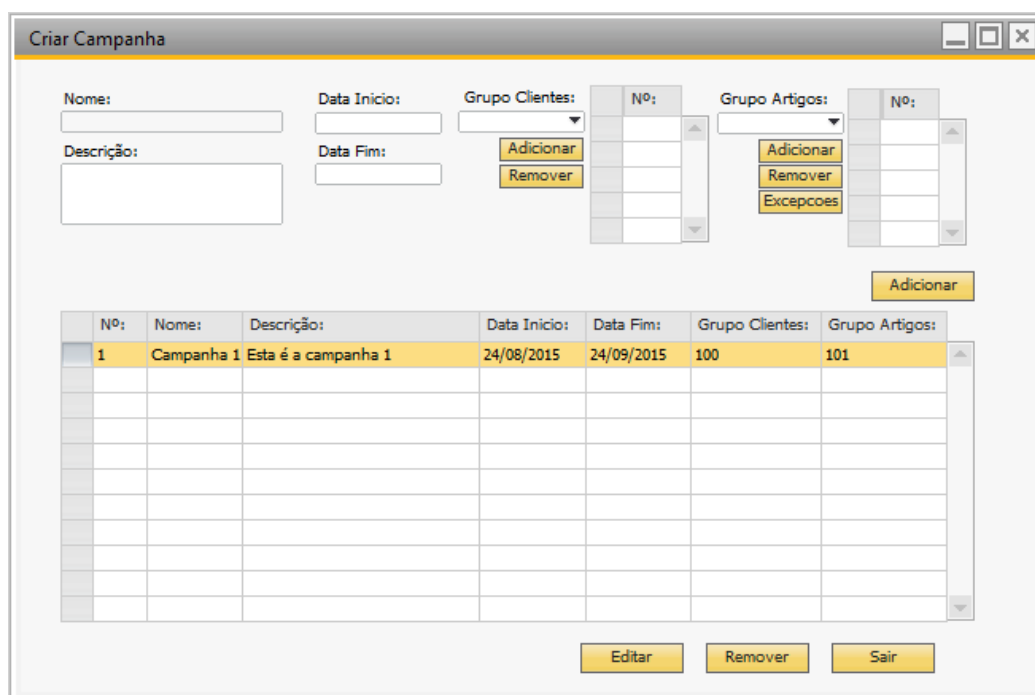


Figura 47: Tutorial menu principal 4

O utilizador selecciona a campanha a ser removida como mostra a Figura 48.

A screenshot of a software window titled "Criar Campanha". The window contains a form with several fields: "Nome:" (text input), "Data Inicio:" (date input), "Data Fim:" (date input), "Grupo Clientes:" (dropdown menu), "Grupo Artigos:" (dropdown menu), and "Descrição:" (text area). There are also two columns of "Nº:" (number) inputs, each with "Adicionar" and "Remover" buttons. Below the form is a table with the following data:

Nº:	Nome:	Descrição:	Data Inicio:	Data Fim:	Grupo Clientes:	Grupo Artigos:
1	Campanha 1	Esta é a campanha 1	24/08/2015	24/09/2015	100	101

At the bottom of the window are three buttons: "Editar", "Remover", and "Sair".

Figura 48: Tutorial seleção de campanha a remover

Após a seleção ter sido realizada e o utilizador ter pressionado o botão ‘Remover’ o sistema deverá apresentar uma mensagem a informar que a

campanha foi removida com sucesso, como mostra a Figura 49.

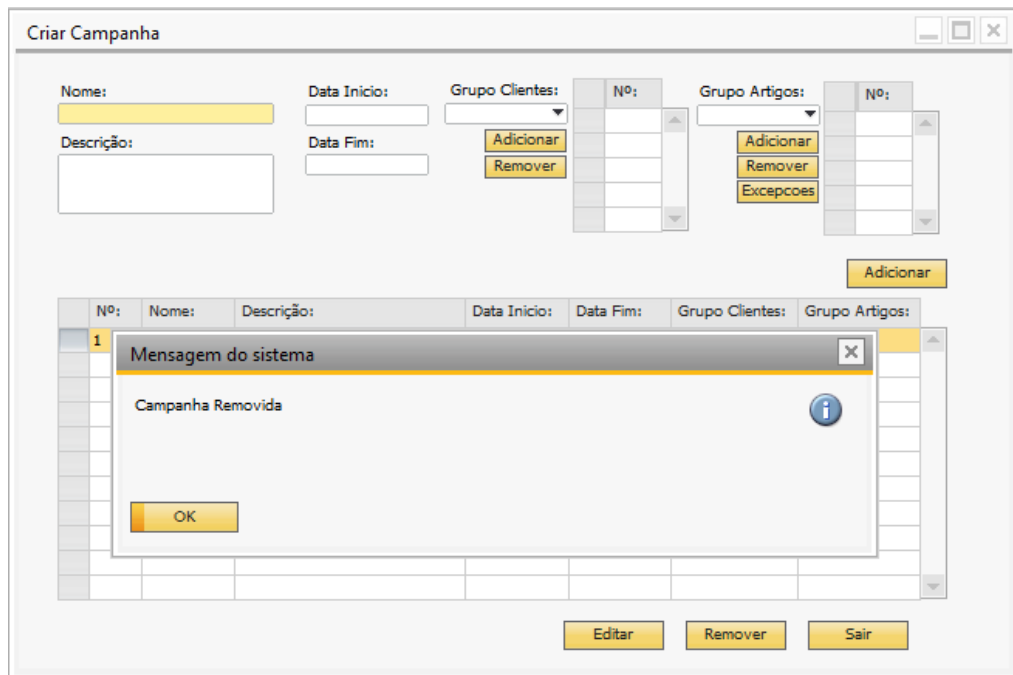


Figura 49: Tutorial remoção completa

A campanha seleccionada será removida da lista como indica a Figura 50.

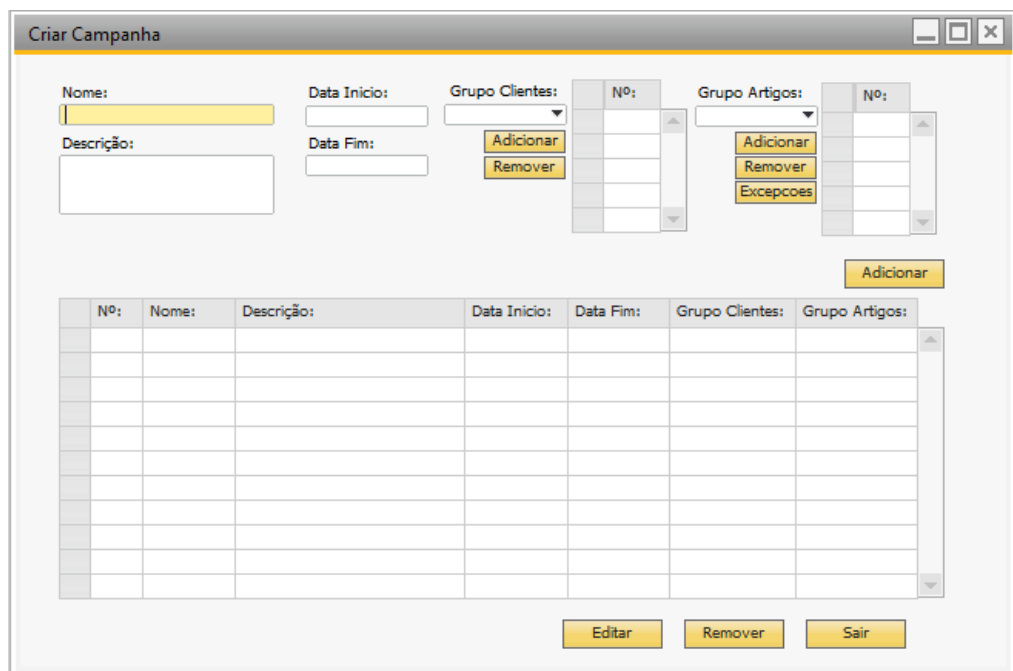


Figura 50: Tutorial remoção completa

## 5.5 Funcionalidade Publicar Campanha

- **Descrição**

A funcionalidade Publicar Campanha deverá ser selecionada no menu principal da aplicação. Aqui o utilizador poderá selecionar a campanha pretendida para ser publicada. Após ter sido selecionada, a campanha deverá ser então enviada para a rede social tendo obrigatoriamente que passar por uma validação de dados. Se o resultado desta validação for positivo a campanha poderá seguir para publicação na rede social.

- **Implementação**

A implementação desta funcionalidade passou por desenvolver uma página específica para a publicação de campanhas. A interface gráfica foi criada no ScreenPainter de forma a respeitar o layout geral do SAP. A página foi desenvolvida de forma a listar todas as campanhas presentes na base de dados. Foi ainda implementado uma caixa de pesquisa para auxiliar o utilizador no caso de um numero elevado de registos na base de dados.

- **Utilização**

O utilizador seleciona a funcionalidade Publicar Campanha do menu principal da aplicação.

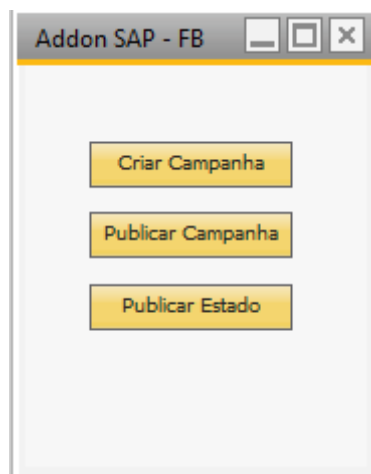


Figura 51: Tutorial menu principal 5

Após selecionar ‘Publicar Campanha’ o utilizador é redirecionado para uma nova página onde seleciona a campanha que pretende publicar da lista como demonstrado na Figura 52.

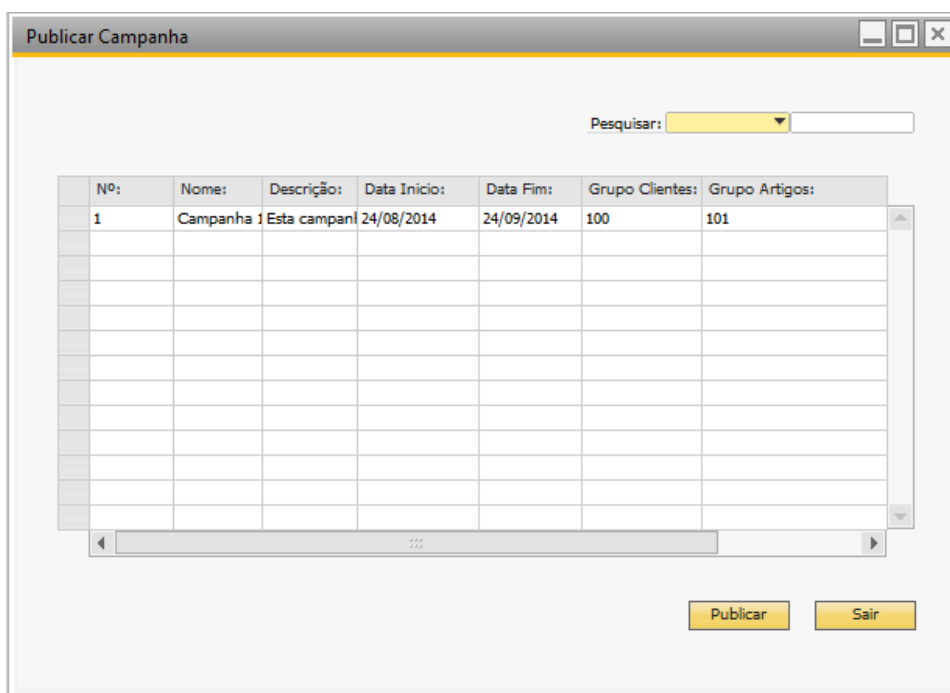


Figura 52: Tutorial listagem publicação campanhas

Após o utilizador pressionar o botão ‘Publicar’ e a campanha ter sido publicada com sucesso, o programa envia uma mensagem de confirmação (Figura 53).

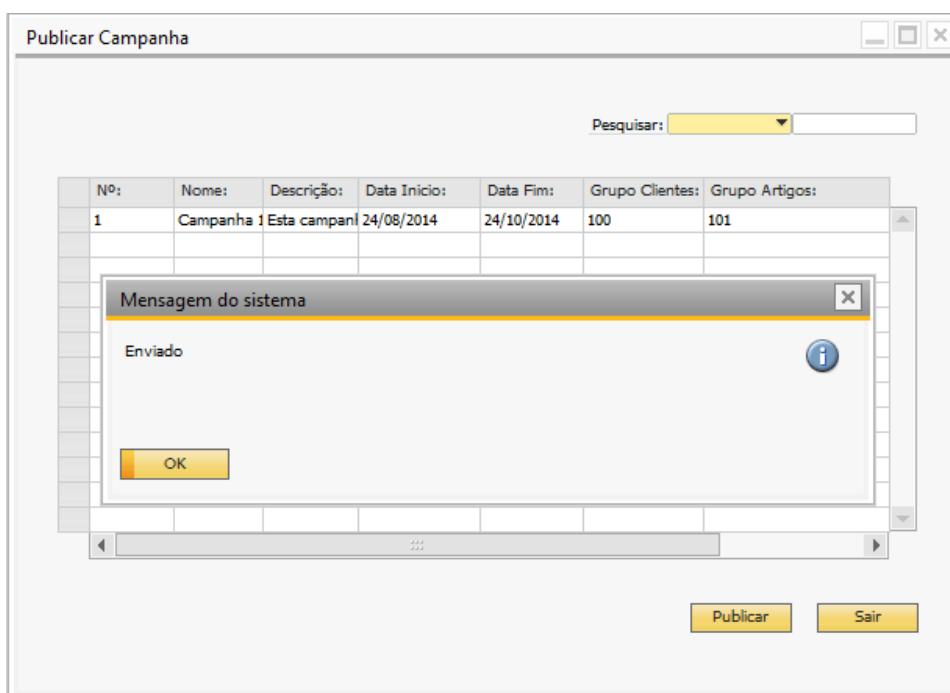


Figura 53: Tutorial publicado com sucesso

Por ultimo é possível ver na Figura 54 o output no Facebook.

Campanha: Campanha 1  
Descrição: Esta campanha vai ser EDITADA.  
Data de Inicio: 2014/08/24  
Data de Fim: 2014/10/24  
Pode encontrar mais informacoes em:  
[www.empresa.pt/campanha/01](http://www.empresa.pt/campanha/01)

Like · Comment

Figura 54: Tutorial output da publicação

## 5.6 Análise e utilização

Os testes efetuados focaram-se nas funcionalidades que o protótipo deveria possuir. Os testes foram efetuados de forma a seguir os passos que um utilizador normalmente seguiria no uso desta aplicação. Isto contribuiu não apenas para a verificação das funcionalidades como permitiu a criação dos tutoriais desenvolvidos nos pontos acima (5.1-5.5).

Tabela 7: Resultado dos testes

Funcionalidade	Descrição	Implementação	Resultado
Login	Função de login para aceder à aplicação	Implementado em C# no SAP e Invocação da API Facebook	Operacional
Criar Campanha	Função de criação de novas campanhas publicitárias	Implementado em C# no SAP	Operacional
Editar Campanha	Função de edição de campanhas previamente criadas	Implementado em C# no SAP	Operacional
Remover Campanha	Função de remoção de campanhas da base de dados	Implementado em C# no SAP	Operacional
Publicar Campanha	Função de publicar a campanha no Facebook	Implementado em C# no SAP e Invocação da API Facebook	Operacional



Através destes testes é possível determinar que todas as funcionalidades do modelo conceptual foram desenvolvidas e implementadas com sucesso no protótipo provando assim a exequibilidade da solução proposta.

Embora não afetando os testes acima realizados ou os seus resultados, foi ainda aplicado uma análise SWOT para melhor entender os pontos fortes e fracos do protótipo e do modelo em si.



Figura 55: Analise SWOT

O modelo está dividido em quatro componentes: forças, fraquezas, oportunidades e ameaças, sendo que existe um relacionamento direto entre forças e oportunidades, fraquezas e ameaças. Nas forças encontra-se a facilidade de publicação de dados diretamente do ERP, a facilidade de acesso já que esta aplicação poderá ser instalada em qualquer terminal que possua o ERP, possui um design simples e intuitivo e ainda tem vantagens financeiras pois a interação com clientes pelas redes sociais tem muito menos gastos que os métodos convencionais. Diretamente relacionado com este componente estão as oportunidades que são a personalização do output publicado no Facebook, podendo ser um pdf, uma página html ou outro tipo a escolha da empresa. A adição de novas funcionalidades como criação de eventos, o aumento da eficácia e eficiência das

próprias funções e ainda a possibilidade de criação de uma plataforma para acesso mobile constituem as restantes oportunidades na análise. As fraquezas consistem no acesso limitado de dados, pois a aplicação apenas consegue aceder a dados do ERP, as limitações impostas pela rede social, e a dependência de funcionamento do SAP. As ameaças ao sistema são a alteração da API da rede social que poderá comprometer as funções atuais, uma mudança significativa na política de addons por parte do SAP e a falta de dados no ERP pode comprometer a eficácia deste sistema.

## 6. Conclusões

O projeto agora desenvolvido tinha como problemática a integração de sistemas de Enterprise Resource Planning e Redes sociais. Este projeto tinha como objetivos a criação de uma proposta de um modelo conceptual que permitisse esta integração e a validação desse modelo através da criação de um protótipo. Para atingir esse fim foi adotada uma metodologia de investigação 'design sciene' ou seja uma metodologia que se baseia em construir e avaliar.

Começou-se por realizar uma revisão compreensiva da literatura sobre a problemática em questão. A partir desta revisão da literatura foi possível entender que os sistemas de informação são sistemas complexos que numa organização funcionam para coleccionar e processar dados para ajudar na tomada de decisão. Foi possível entender a complexidade da integração destes sistemas através de potenciais formas de realizar estas integrações propostas por autores como Hasselbring ou Chari & Seshadri. No entanto esta integração varia de caso para caso e para isso foi necessário também investigar o tipo de SI de interesse no âmbito deste projeto (ERP). Por ultimo a revisão da literatura foi focada no outro sistema, ou seja as redes sociais e através da pesquisa realizada foi possível entender a importância destes sistemas no contexto atual das empresas e CRM permitindo entender ainda a relação completa que ERP, CRM e Redes Sociais possuem.

Após a revisão dos conceitos estar efetuada e seguindo a metodologia proposta, foi criado um modelo conceptual que relacionasse um ERP com uma rede social. Este modelo consistiu em necessidades reais e foi desenvolvido baseado numa framework de integração de sistemas investigada na revisão da literatura tendo sido complementado utilizando UML. Foram explicadas e desenvolvidas as funcionalidades do modelo, através de Use Cases, Diagramas de Classes e Diagramas de Atividades de forma aos requisitos necessários serem perceptíveis para desenvolvimento informático.

Para garantir o segundo objetivo deste projeto e verificar a exequibilidade do modelo atrás criado, foi desenvolvido um protótipo onde se selecionou um ERP e uma Rede Social especifica (SAP e Facebook respetivamente). Ao longo deste capítulo foi explicado como foi realizado o desenvolvimento deste protótipo, desde dos seus detalhes gráficos, desenvolvimento de todas as funcionalidades propostas, até a ligação com o Facebook.

Por fim foram realizados testes funcionais para garantir, em linha com a metodologia aplicada, que o protótipo estava funcional. Através destes testes é possível concluir que o protótipo prova a exequibilidade do modelo, confirmando assim, que os objetivos iniciais para este projeto, foram atingidos.

Após garantir que o protótipo estava operacional e que os objetivos do projeto tinham sido atingidos foi ainda efetuada uma análise swot para um melhor entendimento dos pontos fortes e fracos e também para analisar quais as oportunidades e ameaças possíveis a esta aplicação.

### **6.1 Trabalho Futuro**

Esta aplicação poderia ser complementada incorporando mais funcionalidades do Facebook como a criação de eventos. Esta funcionalidade poderá permitir ao utilizador da aplicação a criação de eventos para os seus clientes como forma de agradecimento ou celebração. Mais, devido a forma como o protótipo foi estruturado, estes eventos poderão ser personalizados da mesma forma que as publicações foram, isto é, criar eventos específicos para cada tipo de cliente.

O output da aplicação poderá ser personalizado a vários níveis. Aqui, um futuro upgrade à aplicação, poderá passar pela criação de um documento .pdf quando a campanha é publicada, permitindo assim aos utilizadores fazerem o download da campanha diretamente do post. Ainda neste ponto, outra alternativa seria gerar uma página html diretamente no site da empresa com a campanha, incluído todos os produtos.

## 7. Bibliografia

Anderson, G. W. 2011. **Sams Teach Yourself SAP**, Fourth Edition

Barr, A 2013. 'What is erp and why is it useful in the custom precast industry?', *Concrete Products*, 116, 9, pp. 42-43, Business Source Complete, EBSCOhost, viewed 16 June 2014.

Bolotaeva, V. & Cata, T. 2010. Marketing Opportunities with Social Networks. **Journal of Internet Social Networking and Virtual Communities** 2010. DOI=10.5171/2010.109111

Boyd, D. M. & Ellison, N. B. 2007. Social Network Sites: Definition, History, and Scholarship. **Journal of Computer - Mediated Communication**, Volume 13, Issue 1, Pages 210–230.

Buttle, F. 2008. **Customer Relationship Management: Concepts and Technologies**, 2nd Edition. Butterworth Heinemann

Chari, K. & Seshadri, S. 2004. Demystifying integration. **Communications of the ACM** 47, 7 (July 2004), 58-63. DOI=10.1145/1005817.1005822

Columbus, L. 2013. 2013 ERP Market Share Update: SAP Solidifies Market Leadership <<http://www.forbes.com/sites/louiscolombus/2013/05/12/2013-erp-market-share-update-sap-solidifies-market-leadership/>> Acedido em Abril 2014

Costa, M.; Resende, R.; Nakamura, E. & Segatto, M. 2008. Software frameworks for information systems integration based on web services. **In Proceedings of the 2008 ACM symposium on Applied computing (SAC '08)**. ACM, New York, NY, USA, 777-782. DOI=10.1145/1363686.1363866

Fayerman, M. 2002. Customer Relationship Management. **New Directions for Institutional Research**, 2002: 57–68. doi: 10.1002/ir.37

Granta, D.; Hwang, Y. & Tuc, Q. 2013. An empirical investigation of six levels of enterprise resource planning integration - **Computers in Human Behavior** Volume 29, Issue 6, Pages 2123-2133

Greco, D. & White, B. 2009. Alphabet soup: How CRM, ERP, Web 2.0 & Sales 2.0 is creating a superior sales experience: A case study. **In: Proc ISECON Proceeding of the Information System Education Conference.** Washington DC, USA.

Hasselbring, W. 2000. Information system integration. **Communications of the ACM** 43, 6 (June 2000), 32-38. DOI=10.1145/336460.336472

Kalus, H., Rosemann, M., Gable, G. 2000. What is ERP? **Information Systems Frontiers** Volume 2, Issue 2, Pages141-162

Kaplan, A. 1964. **The Conduct of Inquiry**, New York: Crowell

Kumar, V. 2010. **Customer Relationship Management.** Wiley International Encyclopedia of Marketing. 1.

Laudon, K. & Laudon, J. 2006. **Management Information Systems: Managing the Digital Firm**, 9Th ed. Prentice Hall

Leon, A. 2008. **Enterprise Resource Planning**, Second Edition

Lepointe, P. 2012. Measuring Facebook's Impact on Marketing, **Journal of Advertising Research** Volume 52, Issue 3, Pages 286-287

Lorette, K. 2014. - What Is the Relationship Between ERP and CRM? <<http://www.wisegeek.com/what-is-the-relationship-between-erp-and-crm.htm>>

Acedido em Setembro de 2014

Stair, R & Reynolds, G. 2013. **Fundamentals of Information Systems**, 7Ed. Cengage Learning

March, S. T. & Smith, G. F. 1995. Design and natural science research on information technology. **Decision Support Systems**, 15(4), pp. 251–266

Pinto, J. P. 2010. **Gestão de operações na indústria e nos serviços.** Editora Lidel

Quast, L. 2012. Recruiting, Reinvented: How Companies Are Using Social Media In The Hiring Process <<http://www.forbes.com/sites/lisaquast/2012/05/21/recruiting-reinvented-how-companies-are-using-social-media-in-the-hiring-process>> Acedido em Abril 2014

Rainer, R. K. & Cegielski, C. G. 2010. **Introduction to Information Systems: Enabling and Transforming Business**, 4Th ed, John Wiley & Sons

Rumbaugh, J.; Jacobson, I. & Booch, G. 2005. **The Unified Modeling Language Reference Manual** , Addison-Wesley , Boston, MA .

Sabooniha, N.; Toohey, D. & Lee, K. 2012. An evaluation of hospital information systems integration approaches. **In Proceedings of the International Conference on Advances in Computing, Communications and Informatics (ICACCI '12)**. ACM, New York, NY, USA, 498-504. DOI=10.1145/2345396.2345479

Schwaiger, W. S. A. & Abmayer, M. 2013. Accounting and Management Information Systems: A Semantic Integration. **In Proceedings of International Conference on Information Integration and Web-based Applications & Services (IIWAS '13)**. ACM, New York, NY, USA, , Pages 346 , 7 pages. DOI=10.1145/2539150.2539214

Sobre o SAP 2013. Disponível em <<http://www.sap.com/about.html>> Acedido em Julho de 2013

The Sage ERP Team. 2011. - Social Media and ERP a helpfull combination <<http://blog.sageerpsolutions.com/social-media-and-erp-a-helpful-combination/>> - Acedido em Março 2014

Weston, R. 2008. 7 Social Networking Strategies. <<http://www.entrepreneur.com/technology/bmighty/article191312.html>> Acedido em Junho 2014

Woodcock, Neil, Green, & Michael Starkey. 2011 - Social CRM as a business strategy. **Journal of Database Marketing & Customer Strategy Management** 18.1: 50-64.

## **8. Anexos**



## Anexo 1 - Código Fonte

```
using System;
using System.Windows.Forms;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Facebook;
using System.Net;
using System.IO;
using System.Dynamic;

namespace Segundo
{
    class Teste
    {
        //App and Form variables
        private SAPbouiCOM.Application SBO_Application;
        private SAPbouiCOM.Form oForm;
        private SAPbouiCOM.Item oItem;
        private SAPbobsCOM.Company oCompany;

        //Botoes
        private SAPbouiCOM.Button oButton1;
        private SAPbouiCOM.Button oButton2;
        private SAPbouiCOM.Button oButton3;
        private SAPbouiCOM.Button oButton4;
        private SAPbouiCOM.Button oButton5;
        private SAPbouiCOM.Button oButton6;

        //ComboBoxes
        private SAPbouiCOM.ComboBox oCombo;
        private SAPbouiCOM.ComboBox oCombo1;

        //CheckBoxs
        private SAPbouiCOM.CheckBox oCheck;
        private SAPbouiCOM.CheckBox oCheck1;
        private SAPbouiCOM.CheckBox oCheck2;
        private SAPbouiCOM.CheckBox oCheck3;
        private SAPbouiCOM.CheckBox oCheck4;

        //StaticText and EditText
        private SAPbouiCOM.EditText oEdit;
        private SAPbouiCOM.EditText oEdit1;
        private SAPbouiCOM.EditText oEdit2;
        private SAPbouiCOM.EditText oLink;
        private SAPbouiCOM.EditText ExtEdit;
        private SAPbouiCOM.StaticText StaText;
        private SAPbouiCOM.StaticText StaText1;
        private SAPbouiCOM.StaticText StaText2;
        private SAPbouiCOM.StaticText StaText3;

        //Option Button
        private SAPbouiCOM.OptionBtn Opt1;
        private SAPbouiCOM.OptionBtn Opt2;

        //User Data Sources
        private SAPbouiCOM.UserDataSource userDB;
        private SAPbouiCOM.UserDataSource userDB1;
        private SAPbouiCOM.UserDataSource userDB2;
        private SAPbouiCOM.UserDataSource userDB3;
```

```

private SAPbouiCOM.UserDataSource userDB4;
private SAPbouiCOM.UserDataSource formDB;

//DB Data Sources
private SAPbouiCOM.DBDataSource oDBDataSource;
private SAPbouiCOM.DBDataSource oDBDataSource1;
private SAPbouiCOM.DBDataSource oDBDSorce;
private SAPbouiCOM.DBDataSource oDBCliente;
private SAPbouiCOM.DBDataSource oDBArtigo;
private SAPbouiCOM.LinkedTextBox oLinked;

//Matrix e Colunas
private SAPbouiCOM.Column oColumn1;
private SAPbouiCOM.Column oColumn2;
private SAPbouiCOM.Column oColumn3;
private SAPbouiCOM.Column oColumn4;
private SAPbouiCOM.Column oColumn5;
private SAPbouiCOM.Column oColumn6;
private SAPbouiCOM.Column oColumn7;
private SAPbouiCOM.Columns oColumns;
private SAPbouiCOM.Matrix oMatrix;
private SAPbouiCOM.Matrix oMatrixCliente;
private SAPbouiCOM.Matrix oMatrixArtigo;
private SAPbouiCOM.Matrix oMatrixExp;

private SAPbobsCOM.Recordset oRecordset = null;
private SAPbobsCOM.Recordset RSO = null;
private SAPbobsCOM.UserTable UT = null, UTKey = null;
private SAPbobsCOM.UserTables UTS = null;
string sSQL = null;

String temp;
String temp2;
String temp3;
String FacebookID;
String ItemID;
String Campanha;
String ArtigosComp;
String campoclientes;
String campoartigos;
int pesquisa = 0, artigos = 0, clientes = 0, posicao = 0, editcampanha, a;
int artexp;
int matrixremove, comboremoverpesquisa = 0, artexpremove, modoeditar;

private void SetApplication()
{
    SAPbouiCOM.SboGuiApi SboGuiApi = null;
    string sConnectionString = null;

    SboGuiApi = new SAPbouiCOM.SboGuiApi();

    sConnectionString = System.Convert.ToString(Environment.GetCommandLineArgs().GetValue(1));

    try
    {
        SboGuiApi.Connect(sConnectionString);
        SBO_Application = SboGuiApi.GetApplication(-1);
    }
    catch
    {
        MessageBox.Show("Connection Failed!");
    }
}
}

```

```

private int SetConnectionContext()
{
    int setConnectionContextReturn = 0;

    string sCookie = null;
    string sConnectionContext = null;
    int lRetCode = 0;

    oCompany = new SAPbobsCOM.Company();

    sCookie = oCompany.GetContextCookie();
    sConnectionContext = SBO_Application.Company.GetConnectionContext(sCookie);

    if (oCompany.Connected == true)
    {
        oCompany.Disconnect();
    }
    setConnectionContextReturn = oCompany.SetSboLoginContext(sConnectionContext);

    return setConnectionContextReturn;
}

private int ConnectToCompany()
{
    int connectToCompanyReturn = 0;
    // Establish the connection to the company database.
    connectToCompanyReturn = oCompany.Connect();

    return connectToCompanyReturn;
}

private void LoadFromXML(ref string FileName)
{
    try
    {
        System.Xml.XmlDocument oXmlDoc = null;

        oXmlDoc = new System.Xml.XmlDocument();

        // load the content of the XML File
        string sPath = null;

        sPath
System.IO.Directory.GetParent(System.IO.Directory.GetParent(Application.StartupPath).ToString()).ToString(); =
        //SBO_Application.MessageBox(sPath, 1, "Ok", "", "");
        oXmlDoc.Load(sPath + @"\" + FileName);

        // load the form to the SBO application in one batch
        string strXML = oXmlDoc.InnerXml.ToString();
        SBO_Application.LoadBatchActions(ref strXML);
    }
    catch
    {
        SBO_Application.MessageBox("Error Loading Form", 1, "Ok", "", "");
    }
}

public Teste()
{

    SetApplication();
    if (!(SetConnectionContext() == 0))

```

```

    {
        SBO_Application.MessageBox("Failed setting a connection to DI API", 1, "Ok", "", "");
        System.Environment.Exit(0); // Terminating the Add-On Application
    }
    if (!(ConnectToCompany() == 0))
    {
        SBO_Application.MessageBox("Failed connecting to the company's Data Base", 1, "Ok", "", "");
        Application.Exit(); // Terminating the Add-On Application
    }

    //SBO_Application.MessageBox("DI Connected To: " + oCompany.CompanyName, 1, "Ok", "", "");

    // Load the the form
    string strTmp = "UITeste0.srf";
    LoadFromXML(ref strTmp);

    // Get the added form object by using the form's UID
    oForm = SBO_Application.Forms.Item("frmPrimeiro");
    // Show the form
    oForm.Visible = true;

    // events handled by SBO_Application_ItemEvent
    SBO_Application.ItemEvent += new
    SAPbouiCOM._IApplicationEvents_ItemEventEventHandler(SBO_Application_ItemEvent);

}

private void SBO_Application_ItemEvent(string FormUID, ref SAPbouiCOM.ItemEvent pVal, out bool
BubbleEvent)
{
    BubbleEvent = true;
    SAPbouiCOM.BoEventTypes EventEnum = 0;
    EventEnum = pVal.EventType;

    #region EventosPublicarCampanha
    if (FormUID == "frmQuarto")
    {
        if (pVal.Before_Action == false)
        {
            if (pVal.ItemUID == "Publicar1" && pVal.EventType ==
SAPbouiCOM.BoEventTypes.et_ITEM_PRESSED)
            {
                CompilarCampanha();
            }

            if (pVal.ItemUID == "Sair1" && pVal.EventType ==
SAPbouiCOM.BoEventTypes.et_ITEM_PRESSED)
            {
                comboremoverpesquisa = 0;
                oForm.Close();
            }
            if (pVal.ItemUID == "Edit1" && pVal.EventType == SAPbouiCOM.BoEventTypes.et_KEY_DOWN)
            {
                if (pVal.CharPressed == 13)
                {
                    string pesquisatemp;
                    string pesquisatipo;
                    pesquisatemp = oEdit.String;
                    try
                    {
                        if (oCombo.Selected.Value.ToString() == "Nome")
                        {
                            pesquisatipo = "Nome";
                            PesquisaRemoverCampanha(pesquisatipo, pesquisatemp);
                        }
                    }
                }
            }
        }
    }
}

```

```

        if (oCombo.Selected.Value.ToString() == "Numero")
        {
            pesquisatipo = "Numero";
            PesquisaRemoverCampanha(pesquisatipo, pesquisatemp);
        }
    }
    catch (NullReferenceException)
    { SBO_Application.MessageBox("Selecione o campo de pesquisa.", 1, "OK", "", ""); }
}
}
}
}
}
#endregion

#region Excepcoes Artigos
if (FormUID == "frmartigos")
{
    if (pVal.ItemUID == "Sair1" && pVal.EventType == SAPbouiCOM.BoEventTypes.et_ITEM_PRESSED)
    {
        oForm.Close();
    }

    if (pVal.ItemUID == "Mudar1" && pVal.EventType ==
SAPbouiCOM.BoEventTypes.et_ITEM_PRESSED)
    {
        MudarEstadoEA();
    }
}
}
#endregion

#region EventosCriarCampanha
if (FormUID == "frmTerceiro")
{
    if (pVal.Before_Action == false)
    {
        if (pVal.ItemUID == "Button1" && pVal.EventType ==
SAPbouiCOM.BoEventTypes.et_ITEM_PRESSED)
        {
            if (oButton2.Caption == "Adicionar")
                AdicionarCampanha();
            else
                EditarValores();
        }

        if (pVal.ItemUID == "Add1" && pVal.EventType ==
SAPbouiCOM.BoEventTypes.et_ITEM_PRESSED)
        {
            if (oButton2.Caption == "Adicionar")
                AdicionarGCliente(posicao);
            else
                AdicionarGCliente(editcampanha);
        }

        if (pVal.ItemUID == "Add2" && pVal.EventType ==
SAPbouiCOM.BoEventTypes.et_ITEM_PRESSED)
        {
            if (oButton2.Caption == "Adicionar")
                AdicionarGArtigo(posicao);
            else

```

```

        AdicionarGArtigo(editcampanha);
    }

    if (pVal.ItemUID == "Rem1" && pVal.EventType ==
SAPbouiCOM.BoEventTypes.et_ITEM_PRESSED)
    {
        RemoverGcliente(posicao);
    }

    if (pVal.ItemUID == "Rem2" && pVal.EventType ==
SAPbouiCOM.BoEventTypes.et_ITEM_PRESSED)
    {
        if (oButton2.Caption == "Adicionar")
            RemoverGartigo(posicao);
        else
        {
            RemoverGartigo(editcampanha);
        }
    }

    if (pVal.ItemUID == "Sair1" && pVal.EventType ==
SAPbouiCOM.BoEventTypes.et_ITEM_PRESSED)
    {
        oForm.Close();
    }

    if (pVal.ItemUID == "Editar1" && pVal.EventType ==
SAPbouiCOM.BoEventTypes.et_ITEM_PRESSED)
    {
        modoeditar = 1;
        EditarCampanha();
    }

    if (pVal.ItemUID == "Remover1" && pVal.EventType ==
SAPbouiCOM.BoEventTypes.et_ITEM_PRESSED)
    {
        RemoverCampanha();
    }

    if (pVal.ItemUID == "Exp1" && pVal.EventType ==
SAPbouiCOM.BoEventTypes.et_ITEM_PRESSED)
    {
        string Tmp = "UITesteArtigo.srf";
        LoadFromXML(ref Tmp);
        LoadVariaveisEA();
    }
}

}

#endregion

#region EventosPublicarEstado
if (FormUID == "frmSegundo")
{
    if (pVal.Before_Action == false)
    {
        if (pVal.ItemUID == "Button1" && pVal.EventType ==
SAPbouiCOM.BoEventTypes.et_ITEM_PRESSED)
        {

            int check1 = 0, check2 = 0, check3 = 0, check4 = 0, check5 = 0;
            int a = 0, b = 0;

            b=LinhaSelecionada();

```

```

if (oCheck.Checked == true)
{
    check1 = 2;
    a++;
}

if (oCheck1.Checked == true)
{
    check2 = 2;
    a++;
}

if (oCheck2.Checked == true)
{
    check3 = 2;
    a++;
}

if (oCheck3.Checked == true)
{
    check4 = 2;
    a++;
}

if (oCheck4.Checked == true)
{
    check5 = 2;
    a++;
}

if (a > 0 && b != 0)
    Compilar(check1, check2, check3, check4, check5);
else
    SBO_Application.MessageBox("Verifique se selecionou os campos e as linhas a exportar.", 1,
"OK", "", "");
}

```

```

if (pVal.ItemUID == "Load1" && pVal.EventType ==
SAPbouiCOM.BoEventTypes.et_ITEM_PRESSED)
{
    if (oCombo.Active)
    {
        if (oCombo.Selected.Value.ToString() == "Clientes")
        {
            PreencherMatrixClientes();
            PreencherComboBoxPesquisa();
        }

        if (oCombo.Selected.Value.ToString() == "Fornecedores")
        {
            PreencherMatrixFornecedores();
            PreencherComboBoxPesquisa();
        }

        if (oCombo.Selected.Value.ToString() == "Lead")
        {
            PreencherMatrixLead();
            PreencherComboBoxPesquisa();
        }

        if (oCombo.Selected.Value.ToString() == "Artigos")
        {
            PreencherMatrixArtigos();
            PreencherComboBoxPesquisaArtigos();
        }
    }
}

```

```

    }
    else SBO_Application.MessageBox("Por favor seleccione uma opção.", 1, "OK", "", "");
}

if (pVal.ItemUID == "EditBox1" && pVal.EventType ==
SAPbouiCOM.BoEventTypes.et_KEY_DOWN)
{
    if (pVal.CharPressed == 13)
    {
        string pesquisatemp;
        string pesquisatipo;
        pesquisatemp = oEdit.String;
        try
        {
            if (oCombo1.Selected.Value.ToString() == "Nome")
            {
                pesquisatipo = "Nome";
                PesquisaClientes(pesquisatemp, pesquisatipo);
            }
            if (oCombo1.Selected.Value.ToString() == "Tipo")
            {
                pesquisatipo = "Tipo";
                PesquisaClientes(pesquisatemp, pesquisatipo);
            }
            if (oCombo1.Selected.Value.ToString() == "WebSite")
            {
                pesquisatipo = "WebSite";
                PesquisaClientes(pesquisatemp, pesquisatipo);
            }
            if (oCombo1.Selected.Value.ToString() == "Morada")
            {
                pesquisatipo = "Morada";
                PesquisaClientes(pesquisatemp, pesquisatipo);
            }
            if (oCombo1.Selected.Value.ToString() == "Stock")
            {
                pesquisatipo = "Stock";
                PesquisaClientes(pesquisatemp, pesquisatipo);
            }
            if (oCombo1.Selected.Value.ToString() == "Preço")
            {
                pesquisatipo = "Preço";
                PesquisaClientes(pesquisatemp, pesquisatipo);
            }
            if (oCombo1.Selected.Value.ToString() == "Código IVA")
            {
                pesquisatipo = "Código IVA";
                PesquisaClientes(pesquisatemp, pesquisatipo);
            }
        }
        catch(NullReferenceException)
        { SBO_Application.MessageBox("Selecione o campo de pesquisa.", 1, "OK", "", ""); }
    }
}

if (pVal.ItemUID == "Sele1" && pVal.EventType ==
SAPbouiCOM.BoEventTypes.et_ITEM_PRESSED)
{
    for (int i = 1; i <= oMatrix.RowCount; i++)
    {
        oMatrix.SelectRow(i, true, true);
    }
}

```



```

    }

    if (pVal.ItemUID == "Limpar1" && pVal.EventType ==
SAPbouiCOM.BoEventTypes.et_ITEM_PRESSED)
    {
        oMatrix.ClearSelections();
    }

    if (pVal.ItemUID == "browse1" && pVal.EventType ==
SAPbouiCOM.BoEventTypes.et_ITEM_PRESSED)
    {
        OpenFileDialog opf = new OpenFileDialog();
        opf.Filter = "JPG Images|*.jpg|PNG Images|*.png";
        if (opf.ShowDialog() == System.Windows.Forms.DialogResult.OK && opf.FileName !=
string.Empty)
        {
            oLink.String = opf.FileName;
        }
    }
}
#endregion

#region Form Inicial
if (FormUID == "frmPrimeiro")
{
    if (pVal.Before_Action == false)
    {
        if (pVal.ItemUID == "Criar3" && pVal.EventType ==
SAPbouiCOM.BoEventTypes.et_ITEM_PRESSED)
        {
            string Tmp = "UITeste.srf";
            LoadFromXML(ref Tmp);
            LoadVariveis();
        }
    }

    if (pVal.Before_Action == false)
    {
        if (pVal.ItemUID == "Criar2" && pVal.EventType ==
SAPbouiCOM.BoEventTypes.et_ITEM_PRESSED)
        {
            string Tmp = "UITeste2.srf";
            LoadFromXML(ref Tmp);
            LoadVariaveisPublicarCampanha();
        }
    }

    if (pVal.Before_Action == false)
    {
        if (pVal.ItemUID == "Criar1" && pVal.EventType ==
SAPbouiCOM.BoEventTypes.et_ITEM_PRESSED)
        {
            string Tmp = "UITeste1.srf";
            LoadFromXML(ref Tmp);
            LoadVariaveisCriar();
        }
    }
}
#endregion

}

#region Exceções Artigos

```

```

private void LoadVariaveisEA()
{
    oForm = SBO_Application.Forms.Item("frmartigos");
    oDBDataSource = oForm.DataSources.DBDataSources.Add("@TESEAA");
    if (modoeditar == 1)
        LoadEAMatrix(editcampanha);
    else
        LoadEAMatrix(posicao);
}

private void LoadEAMatrix(int posicao)
{
    oItem = oForm.Items.Item("matea");
    oMatrixExp = ((SAPbouiCOM.Matrix)(oItem.Specific));
    oColumns = oMatrixExp.Columns;
    oMatrixExp.Clear();

    oColumn1 = oColumns.Item("V_3");
    oColumn1.TitleObject.Caption = "Cod:";
    oColumn1.DataBind.SetBound(true, "@TESEAA", "U_artigo");

    oColumn1 = oColumns.Item("V_2");
    oColumn1.TitleObject.Caption = "Nome:";
    oColumn1.DataBind.SetBound(true, "@TESEAA", "U_nome");
    //oLinked = (SAPbouiCOM.LinkButton)oMatrixExp.Columns.Item("ItemCode").ExtendedObject;
    //oLinked.LinkedObject = SAPbouiCOM.BoLinkedObject.If_Items;

    oColumn1 = oColumns.Item("V_1");
    oColumn1.TitleObject.Caption = "Nº Camp:";
    oColumn1.DataBind.SetBound(true, "@TESEAA", "U_campanha");

    oColumn1 = oColumns.Item("V_0");
    oColumn1.TitleObject.Caption = "Grp Art:";
    oColumn1.DataBind.SetBound(true, "@TESEAA", "U_gartigo");

    oColumn1 = oColumns.Item("V_4");
    oColumn1.TitleObject.Caption = "Estado:";
    oColumn1.DataBind.SetBound(true, "@TESEAA", "U_estado");

    oColumn1 = oColumns.Item("V_5");
    oColumn1.TitleObject.Caption = "Nº:";
    oColumn1.DataBind.SetBound(true, "@TESEAA", "U_numero");

    SAPbobsCOM.Recordset RS;

    RS = oCompany.GetBusinessObject(SAPbobsCOM.BoObjectTypes.BoRecordset);

    if (posicao == 0)
        RS.DoQuery("SELECT U_artigo, U_estado, U_gartigo, U_nome, U_numero, U_campanha FROM
dbo.[@TESEAA] WHERE U_campanha =" + artexp + " ORDER BY U_numero");
    else
        RS.DoQuery("SELECT U_artigo, U_estado, U_gartigo, U_nome, U_numero, U_campanha FROM
dbo.[@TESEAA] WHERE U_campanha =" + posicao + " ORDER BY U_numero");

    RS.MoveFirst();
    for (int h = 1; h <= RS.RecordCount; h++)
    {
        if (RS.EoF == false)
        {
            oMatrixExp.AddRow();
            ((SAPbouiCOM.EditText)oMatrixExp.Columns.Item("V_0").Cells.Item(h).Specific).Value =
RS.Fields.Item("U_gartigo").Value;
            ((SAPbouiCOM.EditText)oMatrixExp.Columns.Item("V_1").Cells.Item(h).Specific).Value =
RS.Fields.Item("U_campanha").Value;

```

```

        ((SAPbouiCOM.EditText)oMatrixExp.Columns.Item("V_2").Cells.Item(h).Specific).Value =
RS.Fields.Item("U_nome").Value;
        ((SAPbouiCOM.EditText)oMatrixExp.Columns.Item("V_3").Cells.Item(h).Specific).Value =
RS.Fields.Item("U_artigo").Value;
        ((SAPbouiCOM.EditText)oMatrixExp.Columns.Item("V_4").Cells.Item(h).Specific).Value =
RS.Fields.Item("U_estado").Value;
        ((SAPbouiCOM.EditText)oMatrixExp.Columns.Item("V_5").Cells.Item(h).Specific).Value =
Convert.ToString(RS.Fields.Item("U_numero").Value);
    }
    RS.MoveNext();
}
modoeditar = 0;
}

private int EscolhaMudar()
{
    int i;
    for (i = 1; i <= oMatrixExp.RowCount; i++)
    {
        if (oMatrixExp.IsRowSelected(i))
            return i;
    }
    return 0;
}

private void MudarEstadoEA()
{
    int i = EscolhaMudar();
    SAPbobsCOM.Recordset RS;

    RS = oCompany.GetBusinessObject(SAPbobsCOM.BoObjectTypes.BoRecordset);

    if (i != 0)
    {
        sSQL = "SELECT U_numero, U_artigo FROM dbo.[@TESEAA] WHERE U_numero = " +
((SAPbouiCOM.EditText)oMatrixExp.Columns.Item("V_5").Cells.Item(i).Specific).Value.ToString() + " ORDER
BY U_numero";

        RS.DoQuery(sSQL);
        RS.MoveFirst();
        while (RS.EoF == false)
        {
            sSQL = System.Convert.ToString(RS.Fields.Item("U_numero").Value);
            UTS = oCompany.UserTables;
            UT = UTS.Item("TESEAA");
            UT.GetByKey(sSQL);
            if (UT.UserFields.Fields.Item("U_estado").Value == "Activado")
            {
                UT.UserFields.Fields.Item("U_estado").Value = "Desativado";
            }
            else
                UT.UserFields.Fields.Item("U_estado").Value = "Activado";

            UT.Update();
            RS.MoveNext();
        }

        if (modoeditar == 1)
            LoadEAMatrix(editcampanha);
        else
            LoadEAMatrix(posicao);
    }
}

```

```

#endregion

#region PublicarCampanha
private void LoadVariaveisPublicarCampanha()
{
    oForm = SBO_Application.Forms.Item("frmQuarto");

    oItem = oForm.Items.Item("Combo1");
    oCombo = ((SAPbouiCOM.ComboBox)(oItem.Specific));

    oItem = oForm.Items.Item("Edit1");
    oEdit = ((SAPbouiCOM.EditText)(oItem.Specific));

    if (comboremovepesquisa == 0)
    {
        PreencherComboRemover();
    }

    oDBDataSource = oForm.DataSources.DBDataSources.Add("@TESE");
    LoadMatrixCampanha();
}
private void LoadMatrixCampanha()
{
    oItem = oForm.Items.Item("matrix2");
    oMatrix = ((SAPbouiCOM.Matrix)(oItem.Specific));
    oColumns = oMatrix.Columns;
    oMatrix.Clear();

    oColumn7 = oColumns.Item("V_9");
    oColumn7.TitleObject.Caption = "Publicado:";
    oColumn7.DataBind.SetBound(true, "@TESE", "U_Publicado");

    oColumn7 = oColumns.Item("V_8");
    oColumn7.TitleObject.Caption = "Estado:";
    oColumn7.DataBind.SetBound(true, "@TESE", "U_Estado");

    oColumn7 = oColumns.Item("V_7");
    oColumn7.TitleObject.Caption = "ID Facebook:";
    oColumn7.DataBind.SetBound(true, "@TESE", "U_Facebookid");

    oColumn1 = oColumns.Item("V_6");
    oColumn1.TitleObject.Caption = "N°:";
    oColumn1.DataBind.SetBound(true, "@TESE", "U_Numero");

    oColumn2 = oColumns.Item("V_5");
    oColumn2.TitleObject.Caption = "Nome:";
    oColumn2.DataBind.SetBound(true, "@TESE", "U_Nome");

    oColumn3 = oColumns.Item("V_4");
    oColumn3.TitleObject.Caption = "Descrição:";
    oColumn3.DataBind.SetBound(true, "@TESE", "U_Descricao");

    oColumn4 = oColumns.Item("V_3");
    oColumn4.TitleObject.Caption = "Data Inicio:";
    oColumn4.DataBind.SetBound(true, "@TESE", "U_Datainicio");

    oColumn5 = oColumns.Item("V_2");
    oColumn5.TitleObject.Caption = "Data Fim:";
    oColumn5.DataBind.SetBound(true, "@TESE", "U_Datafim");

    oColumn6 = oColumns.Item("V_1");
    oColumn6.TitleObject.Caption = "Grupo Clientes:";
    oColumn6.DataBind.SetBound(true, "@TESE", "U_Gclientes");

    oColumn7 = oColumns.Item("V_0");
    oColumn7.TitleObject.Caption = "Grupo Artigos:";
}

```

```

oColumn7.DataBind.SetBound(true, "@TESE", "U_Gartigos");

oDBDataSource.Query(null);
oMatrix.LoadFromDataSource();
}
private int CampanhaSelecionada()
{
    for (int i = 1; i <= oMatrix.RowCount; i++)
    {
        if (oMatrix.IsRowSelected(i))
            return i;
    }

    return 0;
}
private void PreencherComboRemove()
{
    oCombo.ValidValues.Add("Numero", "Numero Campanha");
    oCombo.ValidValues.Add("Nome", "Nome Campanha");
    comboremovepesquisa = 1;
}
private void PesquisaRemoveCampanha(string tipo, string pesquisa)
{
    oMatrix.ClearSelections();
    if (tipo == "Nome")
    {
        for (int i = 1; i <= oMatrix.RowCount; i++)
        {
            if (((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_5").Cells.Item(i).Specific).Value.ToString() ==
pesquisa)
            {
                oMatrix.SelectRow(i, true, true);
            }
        }
    }

    if (tipo == "Numero")
    {
        for (int i = 1; i <= oMatrix.RowCount; i++)
        {
            if (((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_6").Cells.Item(i).Specific).Value.ToString() ==
pesquisa)
            {
                oMatrix.SelectRow(i, true, true);
            }
        }
    }
}
private int Data(string lol)
{
    string a, b = "";
    a = lol;
    b = a.Substring(6, 2);
    b += "/";
    b += a.Substring(4, 2);
    b += "/";
    b += a.Substring(0, 4);
    DateTime dt1 = DateTime.Parse(b);
    DateTime dt2 = DateTime.Now;

    if (dt1.Date < dt2.Date)
    {
        return 0;
    }
    else
    {

```

```

        return 1;
    }
}
private void CompilarCampanha()
{
    int i;
    RSO
((SAPbobsCOM.Recordset)oCompany.GetBusinessObject(SAPbobsCOM.BoObjectTypes.BoRecordset));
    matrixremover = CampanhaSeleccionada();
    i = matrixremover;
    a = Data(((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_2").Cells.Item(i).Specific).Value.ToString());
    LoadMatrixCampanha();

    SAPbobsCOM.Recordset RS;

    RS = oCompany.GetBusinessObject(SAPbobsCOM.BoObjectTypes.BoRecordset);

    sSQL = "SELECT * FROM dbo.[@TESEAA] WHERE U_campanha = " +
((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_6").Cells.Item(i).Specific).Value.ToString() + " AND U_estado
= 'Activado'";

    RS.DoQuery(sSQL);

    SBO_Application.MessageBox(a.ToString(), 1, "OK", "", "");
    if (RS.RecordCount == 1 && a == 1)
    {
        RS.MoveFirst();
        while (RS.EoF == false)
        {
            ArtigosComp += System.Convert.ToString(RS.Fields.Item("U_nome").Value);
            ArtigosComp += "\n";
            UTS = oCompany.UserTables;
            UT = UTS.Item("TESEAA");
            UT.GetByKey(sSQL);
            UT.Remove();
            RS.MoveNext();
        }
    }
    else if (((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_9").Cells.Item(i).Specific).Value.ToString() ==
"0" && RS.RecordCount > 1 && a == 1)
    {
        ItemID
((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_0").Cells.Item(i).Specific).Value.ToString();
        FacebookID
((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_7").Cells.Item(i).Specific).Value.ToString();

        Campanha = "Campanha: ";
        Campanha
((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_5").Cells.Item(i).Specific).Value.ToString();
        Campanha += "\n";
        Campanha += "Descrição: ";
        Campanha
((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_4").Cells.Item(i).Specific).Value.ToString();
        Campanha += "\n";
        Campanha += "Data de Inicio: ";
        Campanha
((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_3").Cells.Item(i).Specific).Value.ToString();
        Campanha += "\n";
        Campanha += "Data de Fim: ";
        Campanha
((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_2").Cells.Item(i).Specific).Value.ToString();
        Campanha += "\n";
        //Campanha += "Grupo de Clientes: ";
        //Campanha
((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_1").Cells.Item(i).Specific).Value.ToString();
        //Campanha += "\n";

```

```

//Campanha += "Grupo de Artigos: ";
//Campanha
((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_0").Cells.Item(i).Specific).Value.ToString();
//Campanha += "\n";

RSO.DoQuery("SELECT U_nome, U_preco, U_numero FROM dbo.[@TESEAA] WHERE U_campanha
= " + ((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_6").Cells.Item(i).Specific).Value.ToString() + """);

RSO.MoveFirst();
ArtigosComp += Campanha;
Campanha += "Pode encontrar mais informacoes em: \n www.empresa.pt/campanha/01";
while (RSO.EoF == false)
{
    sSQL = System.Convert.ToString(RSO.Fields.Item("U_numero").Value);
    UTS = oCompany.UserTables;
    UT = UTS.Item("TESEAA");
    UT.GetKey(sSQL);
    ArtigosComp += System.Convert.ToString(RSO.Fields.Item("U_nome").Value);
    ArtigosComp += "\n";
    ArtigosComp += System.Convert.ToString(RSO.Fields.Item("U_preco").Value);
    ArtigosComp += "\n";
    RSO.MoveNext();
}

string campanha1 = Campanha; // impede a perda do valor da variavel por um motivo que eu apos 3 horas a
tentar descobrir ainda desconheço
RS = oCompany.GetBusinessObject(SAPbobsCOM.BoObjectTypes.BoRecordset);

RS.DoQuery("SELECT U_Numero, U_grupo, U_campanha FROM dbo.[@TESEGC] WHERE
U_campanha = """);
((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_6").Cells.Item(i).Specific).Value.ToString()+ """);
RS.MoveFirst();
while (RS.EoF == false)
{
    if (RS.Fields.Item("U_grupo").Value == "100")
    {
        SBO_Application.MessageBox(Campanha, 1, "OK", "", "");
        PublicarCampanha(Campanha, "1459159980984058");
    }
    if (RS.Fields.Item("U_grupo").Value == "102")
    {
        SBO_Application.MessageBox(campanha1, 1, "OK", "", "");
        PublicarCampanha(campanha1, "621821777900150");
    }

    Campanha = "";
    ArtigosComp = "";

    RS.MoveNext();
}

}
else if (a == 0)
{
    SBO_Application.MessageBox("Erro! Campanha já passou a data final.", 1, "OK", "", "");
}
else
{
    SBO_Application.MessageBox("Erro! Campanha já publicada.", 1, "OK", "", "");
}
}
private void PublicarCampanha(string enviar, string grupo)
{
    oRecordset
((SAPbobsCOM.Recordset)(oCompany.GetBusinessObject(SAPbobsCOM.BoObjectTypes.BoRecordset)));

```

```

string token =
"CAACEdEose0cBALdPGVJda5TYXIjwUnR8iboX2XcdChmGkbDIVqpvtovV5ZAH8gVSlmnO3xUwS7FeA5Rh9K
ZBKXYF9TSecXXQHFHSEHZBVUR6ZBdmZBErRg0P0yGMOuTgBNtLHRBx9zgzrtLBjHZCEO8MBAq791X
6Sfh3ECMOkVWwSQ17AQ2aS6H1B4hzZBXfcw6QwoB7ueeCE3OJtbYMc";/"CAATu9oYF1FsBAOUqXsB6gG
0RpkO7ppZCBcygSJYxyZAofctNE7z7S5ZCzpguQlswlhtpC8BFJFooChRyV50zfaZBX51NEc6E7dP2RhIrkHdZCz
aypMmAaHZBuNeB9rYxIUulzmDDUNHtPXXfssr8VlcsIXeOe21lmH3lgJtOTD0F01wFVDS0bF";

FacebookAPI api = new Facebook.FacebookAPI(token);
myToken.Default.userId = grupo.ToString();
Dictionary<string, string> postArgs = new Dictionary<string, string>();
postArgs["message"] = enviar;
JSONObject post = api.Post("/" + grupo.ToString() + "/feed", postArgs);
//JSONObject post = api.Post("/1388642484737115/groups", postArgs);

ItemID = "";

sSQL = "SELECT U_Numero, U_Publicado FROM dbo.[@TESE] WHERE U_Numero = " +
((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_6").Cells.Item(matrixremover).Specific).Value.ToString() +
""";
oRecordset.DoQuery(sSQL);
// Check if a record exists with this [ItemCode] and [SuppSerial]
if (oRecordset.RecordCount > 0)
{
oRecordset.MoveFirst();

sSQL = System.Convert.ToString(oRecordset.Fields.Item("U_Numero").Value);

UTS = oCompany.UserTables;
UT = UTS.Item("TESE");
UT.GetByKey(sSQL);
UT.UserFields.Fields.Item("U_Publicado").Value = "1";
UT.Update();
}
SBO_Application.MessageBox("Enviado", 1, "OK", "", "");
}

#endregion

#region CriarCampanha
private void LoadVariaveisCriar()
{
oForm = SBO_Application.Forms.Item("frmTerceiro");

//EditBox do Texto Extra
oItem = oForm.Items.Item("ExtEdit1");
ExtEdit = oItem.Specific;

oItem = oForm.Items.Item("ComboBox1");
oCombo = ((SAPbouiCOM.ComboBox)oItem.Specific);
addGclientes(oCombo);
oItem = oForm.Items.Item("ComboBox2");
oCombo1 = ((SAPbouiCOM.ComboBox)oItem.Specific);
addGartigos(oCombo1);

//Botão do Load
oItem = oForm.Items.Item("Editar1");
oButton1 = oItem.Specific;

oItem = oForm.Items.Item("Button1");
oButton2 = oItem.Specific;

//EditBox da Pesquisa
oItem = oForm.Items.Item("EditBox1");
oEdit = ((SAPbouiCOM.EditText)(oItem.Specific));
//EditBox da Pesquisa
oItem = oForm.Items.Item("EditBox2");
oEdit1 = ((SAPbouiCOM.EditText)(oItem.Specific));
//EditBox da Pesquisa

```



```

oItem = oForm.Items.Item("EditBox3");
oEdit2 = ((SAPbouiCOM.EditText)(oItem.Specific));

oDBDataSource = oForm.DataSources.DBDataSources.Add("@TESE");
oDBDataSource1 = oForm.DataSources.DBDataSources.Add("@TESEAA");
oDBCliente = oForm.DataSources.DBDataSources.Add("@TESEGC");
oDBArtigo = oForm.DataSources.DBDataSources.Add("@TESEGA");

oItem = oForm.Items.Item("matrixA");
oMatrixArtigo = ((SAPbouiCOM.Matrix)(oItem.Specific));
oColumns = oMatrixArtigo.Columns;
oMatrixArtigo.Clear();

oColumn1 = oColumns.Item("V_0");
oColumn1.TitleObject.Caption = "N°:";
oColumn1.DataBind.SetBound(true, "@TESEGA", "U_grupo");

oColumn1 = oColumns.Item("V_1");
oColumn1.TitleObject.Caption = "N°:";
oColumn1.DataBind.SetBound(true, "@TESEGA", "U_numero");

oColumn1 = oColumns.Item("V_2");
oColumn1.TitleObject.Caption = "N°:";
oColumn1.DataBind.SetBound(true, "@TESEGA", "U_campanha");

oItem = oForm.Items.Item("matrixC");
oMatrixCliente = ((SAPbouiCOM.Matrix)(oItem.Specific));
oColumns = oMatrixCliente.Columns;
oMatrixCliente.Clear();

oColumn1 = oColumns.Item("V_0");
oColumn1.TitleObject.Caption = "N°:";
oColumn1.DataBind.SetBound(true, "@TESEGC", "U_grupo");

oColumn1 = oColumns.Item("V_1");
oColumn1.TitleObject.Caption = "N°:";
oColumn1.DataBind.SetBound(true, "@TESEGA", "U_numero");

LoadMatrix();
}

private void addGclientes(SAPbouiCOM.ComboBox oCombo)
{
    SAPbobsCOM.Recordset RS;

    RS = oCompany.GetBusinessObject(SAPbobsCOM.BoObjectTypes.BoRecordset);

    RS.DoQuery("SELECT GroupCode,GroupName FROM OCRG ORDER BY GroupCode");

    RS.MoveFirst();
    while (RS.EoF == false)
    {
        oCombo.ValidValues.Add(RS.Fields.Item("GroupCode").Value, RS.Fields.Item("GroupName").Value);
        RS.MoveNext();
    }
}

private void addGartigos(SAPbouiCOM.ComboBox oCombo)
{
    SAPbobsCOM.Recordset RS;

    RS = oCompany.GetBusinessObject(SAPbobsCOM.BoObjectTypes.BoRecordset);

    RS.DoQuery("SELECT ItmsGrpCod,ItmsGrpNam FROM OITB ORDER BY ItmsGrpCod");

    RS.MoveFirst();
    while (RS.EoF == false)

```

```

    {
        oCombo.ValidValues.Add(RS.Fields.Item("ItmsGrpCod").Value, RS.Fields.Item("ItmsGrpNam").Value);
        RS.MoveNext();
    }
}

private void LoadMatrix()
{
    oForm = SBO_Application.Forms.Item("frmTerceiro");
    oDBDataSource = oForm.DataSources.DBDataSources.Add("@TESE");
    oItem = oForm.Items.Item("matrix1");
    oMatrix = ((SAPbouiCOM.Matrix)(oItem.Specific));
    oColumns = oMatrix.Columns;
    oMatrix.Clear();

    oColumn1 = oColumns.Item("V_0");
    oColumn1.TitleObject.Caption = "N°:";
    oColumn1.DataBind.SetBound(true, "@TESE", "U_Numero");

    oColumn2 = oColumns.Item("V_6");
    oColumn2.TitleObject.Caption = "Nome:";
    oColumn2.DataBind.SetBound(true, "@TESE", "U_Nome");

    oColumn3 = oColumns.Item("V_5");
    oColumn3.TitleObject.Caption = "Descrição:";
    oColumn3.DataBind.SetBound(true, "@TESE", "U_Descricao");

    oColumn4 = oColumns.Item("V_4");
    oColumn4.TitleObject.Caption = "Data Inicio:";
    oColumn4.DataBind.SetBound(true, "@TESE", "U_Datainicio");

    oColumn5 = oColumns.Item("V_3");
    oColumn5.TitleObject.Caption = "Data Fim:";
    oColumn5.DataBind.SetBound(true, "@TESE", "U_Datafim");

    oColumn6 = oColumns.Item("V_2");
    oColumn6.TitleObject.Caption = "Grupo Clientes:";
    oColumn6.DataBind.SetBound(true, "@TESE", "U_Gclientes");

    oColumn7 = oColumns.Item("V_1");
    oColumn7.TitleObject.Caption = "Grupo Artigos:";
    oColumn7.DataBind.SetBound(true, "@TESE", "U_Gartigos");

    oDBDataSource.Query(null);
    oMatrix.LoadFromDataSource();
}

private void LoadMClientes(int posicao)
{
    oItem = oForm.Items.Item("matrixC");
    oMatrixCliente = ((SAPbouiCOM.Matrix)(oItem.Specific));
    oColumns = oMatrixCliente.Columns;
    oMatrixCliente.Clear();

    oColumn1 = oColumns.Item("V_0");
    oColumn1.TitleObject.Caption = "N°:";
    oColumn1.DataBind.SetBound(true, "@TESEGC", "U_grupo");

    oColumn1 = oColumns.Item("V_1");
    oColumn1.TitleObject.Caption = "N°:";
    oColumn1.DataBind.SetBound(true, "@TESEGA", "U_numero");

    SAPbobsCOM.Recordset RS;

    RS = oCompany.GetBusinessObject(SAPbobsCOM.BoObjectTypes.BoRecordset);

```

```

RS.DoQuery("SELECT U_Numero FROM dbo.[@TESE]");
int k = RS.RecordCount;
k++;

if (posicao == 0)
RS.DoQuery("SELECT U_grupo, U_numero FROM dbo.[@TESEGC] WHERE U_campanha ="+k+"");
else
RS.DoQuery("SELECT U_grupo, U_numero FROM dbo.[@TESEGC] WHERE U_campanha =" + posicao
+ "");

    RS.MoveFirst();
    for (int h = 1; h <= RS.RecordCount; h++)
    {
        if (RS.EoF == false)
        {
            oMatrixCliente.AddRow();
            ((SAPbouiCOM.EditText)oMatrixCliente.Columns.Item("V_0").Cells.Item(h).Specific).Value =
RS.Fields.Item("U_grupo").Value;
            ((SAPbouiCOM.EditText)oMatrixCliente.Columns.Item("V_1").Cells.Item(h).Specific).Value =
RS.Fields.Item("U_numero").Value;
        }
        RS.MoveNext();
    }

}

private void LoadMArtigos(int posicao)
{
    oItem = oForm.Items.Item("matrixA");
    oMatrixArtigo = ((SAPbouiCOM.Matrix)(oItem.Specific));
    oColumns = oMatrixArtigo.Columns;
    oMatrixArtigo.Clear();

    oColumn1 = oColumns.Item("V_0");
    oColumn1.TitleObject.Caption = "N°:";
    oColumn1.DataBind.SetBound(true, "@TESEGA", "U_grupo");

    oColumn1 = oColumns.Item("V_1");
    oColumn1.TitleObject.Caption = "N°:";
    oColumn1.DataBind.SetBound(true, "@TESEGA", "U_numero");

    oColumn1 = oColumns.Item("V_2");
    oColumn1.TitleObject.Caption = "N°:";
    oColumn1.DataBind.SetBound(true, "@TESEGA", "U_campanha");

    SAPbobsCOM.Recordset RS;

    RS = oCompany.GetBusinessObject(SAPbobsCOM.BoObjectTypes.BoRecordset);

    RS.DoQuery("SELECT U_Numero FROM dbo.[@TESE]");
    int k = RS.RecordCount;
    k++;

    if (posicao == 0)
    RS.DoQuery("SELECT U_grupo, U_numero FROM dbo.[@TESEGA] WHERE U_campanha =" + k + "");
    else
    RS.DoQuery("SELECT U_grupo, U_numero FROM dbo.[@TESEGA] WHERE U_campanha =" + posicao
+ "");

    RS.MoveFirst();
    for (int h = 1; h <= RS.RecordCount; h++)
    {
        if (RS.EoF == false)
        {
            oMatrixArtigo.AddRow();
            ((SAPbouiCOM.EditText)oMatrixArtigo.Columns.Item("V_0").Cells.Item(h).Specific).Value =
RS.Fields.Item("U_grupo").Value;

```

```

        ((SAPbouiCOM.EditText)oMatrixArtigo.Columns.Item("V_1").Cells.Item(h).Specific).Value =
RS.Fields.Item("U_numero").Value;
        ((SAPbouiCOM.EditText)oMatrixArtigo.Columns.Item("V_2").Cells.Item(h).Specific).Value =
k.ToString();
    }
    RS.MoveNext();
}
}

private void AdicionarCampanha()
{
    SAPbobsCOM.Recordset RS;
    int i;
    RS = oCompany.GetBusinessObject(SAPbobsCOM.BoObjectTypes.BoRecordset);
    RS.DoQuery("SELECT U_numero FROM dbo.[@TESE] ORDER BY U_numero");
    if (RS.RecordCount > 0)
    {
        RS.MoveLast();
        i = RS.Fields.Item("U_numero").Value;
        i++;
    }
    else
        i = 1;

    UTS = oCompany.UserTables;
    UT = UTS.Item("TESE");
    UT.Code = System.Convert.ToString(i);
    UT.Name = oEdit.String;
    UT.UserFields.Fields.Item("U_Numero").Value = i;
    UT.UserFields.Fields.Item("U_Nome").Value = oEdit.String;
    UT.UserFields.Fields.Item("U_Descricao").Value = ExtEdit.String;
    UT.UserFields.Fields.Item("U_Datainicio").Value = oEdit1.String;
    UT.UserFields.Fields.Item("U_Datafim").Value = oEdit2.String;
    UT.UserFields.Fields.Item("U_Gclientes").Value = campoclientes;
    UT.UserFields.Fields.Item("U_Gartigos").Value = campoartigos;

    if (oCombo.Selected.Value.ToString() == "100")
    {
        UT.UserFields.Fields.Item("U_Facebookid").Value = "1459159980984058";
    }
    else if (oCombo.Selected.Value.ToString() == "102")
    {
        UT.UserFields.Fields.Item("U_Facebookid").Value = "621821777900150";
    }
    else
    {
        UT.UserFields.Fields.Item("U_Estado").Value = "0";
        SBO_Application.MessageBox("AVISO: Grupo de clientes selecionado não possui Facebook", 1, "OK",
"", "");
    }

    UT.Add();

    ExtEdit.String = "";
    oEdit1.String = "";
    oEdit2.String = "";
    oEdit.String = "";
    oMatrixArtigo.Clear();
    oMatrixCliente.Clear();
    campoclientes = "";
    campoartigos = "";

    LoadMatrix();
}

private void AdicionarGCliente(int posicao)
{

```

```

SAPbobsCOM.Recordset RS;
int j;
int k;
RS = oCompany.GetBusinessObject(SAPbobsCOM.BoObjectTypes.BoRecordset);
RS.DoQuery("SELECT U_numero FROM dbo.[@TESEGC]");
RS.MoveLast();
if (RS.RecordCount > 0)
{
    RS.MoveLast();
    string lol = RS.Fields.Item("U_numero").Value;
    j = Convert.ToInt32(lol);
    j++;
}
else
    j = 1;

RS = oCompany.GetBusinessObject(SAPbobsCOM.BoObjectTypes.BoRecordset);
if (posicao == 0)
{
    RS.DoQuery("SELECT U_Numero FROM dbo.[@TESE]");
    k = RS.RecordCount;
    k++;
}
else
    k = posicao;

UTS = oCompany.UserTables;
UT = UTS.Item("TESEGC");
UT.Code = System.Convert.ToString(j);
UT.Name = System.Convert.ToString(j);
UT.UserFields.Fields.Item("U_numero").Value = System.Convert.ToString(j);
UT.UserFields.Fields.Item("U_campanha").Value = System.Convert.ToString(k);
UT.UserFields.Fields.Item("U_grupo").Value = oCombo.Selected.Value.ToString();
campoclientes += oCombo.Selected.Value.ToString();
campoclientes += " ";
UT.Add();
LoadMClientes(posicao);
}

private void AdicionarGArtigo(int posicao)
{
    SAPbobsCOM.Recordset RS;
    int j;
    int k;
    RS = oCompany.GetBusinessObject(SAPbobsCOM.BoObjectTypes.BoRecordset);
    RS.DoQuery("SELECT U_numero FROM dbo.[@TESEGA]");
    if (RS.RecordCount > 0)
    {
        RS.MoveLast();
        string lol = RS.Fields.Item("U_numero").Value;
        j = Convert.ToInt32(lol);
        j++;
    }
    else
        j = 1;

    RS = oCompany.GetBusinessObject(SAPbobsCOM.BoObjectTypes.BoRecordset);
    if (posicao == 0)
    {
        RS.DoQuery("SELECT U_Numero FROM dbo.[@TESE]");
        k = RS.RecordCount;
        k++;
    }
    else
        k = posicao;

    artexp = k; //variavel global para o load das exceções no outro form

```

```

    UTS = oCompany.UserTables;
    UT = UTS.Item("TESEGA");
    UT.Code = System.Convert.ToString(j);
    UT.Name = System.Convert.ToString(j);
    UT.UserFields.Fields.Item("U_numero").Value = System.Convert.ToString(j);
    UT.UserFields.Fields.Item("U_campanha").Value = System.Convert.ToString(k);
    UT.UserFields.Fields.Item("U_grupo").Value = oCombo1.Selected.Value.ToString();
    UT.Add();
    campoartigos += oCombo1.Selected.Value.ToString();
    campoartigos += " ";
    AddTabelaExcepcoes(oCombo1.Selected.Value.ToString(), k);
    LoadMArtigos(posicao);
}

private void AddTabelaExcepcoes(string exp, int camp)
{
    SAPbobsCOM.Recordset RS;
    int j;
    RS = oCompany.GetBusinessObject(SAPbobsCOM.BoObjectTypes.BoRecordset);
    RS.DoQuery("SELECT U_numero FROM dbo.[@TESEAA] ORDER BY U_numero");
    if (RS.RecordCount > 0)
    {
        RS.MoveLast();
        j = RS.Fields.Item("U_numero").Value;
        j++;
    }
    else
        j = 1;

    RS.DoQuery("SELECT ItemName, ItemCode, ItmsGrpCod, LastPurPrc FROM OITM WHERE ItmsGrpCod
= " + exp + "");
    RS.MoveFirst();
    while (RS.EOF == false)
    {
        UTS = oCompany.UserTables;
        UT = UTS.Item("TESEAA");
        UT.Code = System.Convert.ToString(j);
        UT.Name = System.Convert.ToString(j);
        UT.UserFields.Fields.Item("U_nome").Value = RS.Fields.Item("ItemName").Value;
        UT.UserFields.Fields.Item("U_artigo").Value = RS.Fields.Item("ItemCode").Value;
        UT.UserFields.Fields.Item("U_preco").Value = Convert.ToString(RS.Fields.Item("LastPurPrc").Value);
        UT.UserFields.Fields.Item("U_numero").Value = j;
        UT.UserFields.Fields.Item("U_campanha").Value = camp.ToString();
        UT.UserFields.Fields.Item("U_gartigo").Value = exp;
        UT.UserFields.Fields.Item("U_estado").Value = "Activado";
        UT.Add();
        j++;
        RS.MoveNext();
    }
}

private void EditarCampanha()
{
    int i, j;
    string lol;
    i = RemoverSelecionada();

    ExtEdit.String
((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_5").Cells.Item(i).Specific).Value.ToString(); =
    oEdit1.String =
((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_4").Cells.Item(i).Specific).Value.ToString(); =
    oEdit2.String =
((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_3").Cells.Item(i).Specific).Value.ToString(); =
    oEdit.String =
((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_6").Cells.Item(i).Specific).Value.ToString();

```

```

lol = ((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_0").Cells.Item(i).Specific).Value.ToString();
j = Convert.ToInt32(lol);
editcampanha = j; //varivel global para saber no update que campanha atualizar
LoadMArtigos(j);
LoadMClientes(j);
campoclientes = "";
oButton1.Item.Enabled = false;
oButton2.Caption = "Gravar";
}

private void EditarValores()
{
    SAPbobsCOM.Recordset RS;

    RS = oCompany.GetBusinessObject(SAPbobsCOM.BoObjectTypes.BoRecordset);

    sSQL = "SELECT * FROM dbo.[@TESE] WHERE U_numero = '" + editcampanha + "' ORDER BY
U_numero";

    RS.DoQuery(sSQL);
    RS.MoveFirst();
    while (RS.EoF == false)
    {

        sSQL = System.Convert.ToString(RS.Fields.Item("U_numero").Value);
        UTS = oCompany.UserTables;
        UT = UTS.Item("TESE");
        UT.GetByKey(sSQL);
        UT.UserFields.Fields.Item("U_descricao").Value = ExtEdit.String;
        UT.UserFields.Fields.Item("U_nome").Value = oEdit.String;
        UT.UserFields.Fields.Item("U_Datainicio").Value = oEdit1.String;
        UT.UserFields.Fields.Item("U_Datafim").Value = oEdit2.String;

        UT.Update();
        RS.MoveNext();
    }
    oButton1.Item.Enabled = true;
    oButton2.Caption = "Adicionar";
    ExtEdit.String = "";
    oEdit1.String = "";
    oEdit2.String = "";
    oEdit.String = "";
    oMatrixArtigo.Clear();
    oMatrixCliente.Clear();
    campoclientes = "";
    LoadMatrix();
}

private void RemoveTabelaExcepcoes(string art, string lol)
{
    SAPbobsCOM.Recordset RS;

    RS = oCompany.GetBusinessObject(SAPbobsCOM.BoObjectTypes.BoRecordset);

    sSQL = "SELECT U_numero FROM dbo.[@TESEAA] WHERE U_gartigo = '" + art + "' AND U_campanha
= '"+ lol +"'";

    RS.DoQuery(sSQL);
    if (RS.RecordCount > 0)
    {
        RS.MoveFirst();
        while (RS.EoF == false)
        {
            sSQL = System.Convert.ToString(RS.Fields.Item("U_numero").Value);
            UTS = oCompany.UserTables;
            UT = UTS.Item("TESEAA");
            UT.GetByKey(sSQL);

```

```

        UT.Remove();
        RS.MoveNext();
    }
}

private void RemoverTabelaExcepcoesTotal(int lol)
{
    SAPbobsCOM.Recordset RS;

    RS = oCompany.GetBusinessObject(SAPbobsCOM.BoObjectTypes.BoRecordset);

    sSQL = "SELECT * FROM dbo.[@TESEAA] WHERE U_campanha = " + lol + """;

    RS.DoQuery(sSQL);
    RS.MoveFirst();
    while (RS.EOF == false)
    {
        sSQL = System.Convert.ToString(RS.Fields.Item("U_numero").Value);
        UTS = oCompany.UserTables;
        UT = UTS.Item("TESEAA");
        UT.GetByKey(sSQL);
        UT.Remove();
        RS.MoveNext();
    }
}

private void RemoverGcliente(int posicao)
{
    int i;
    for (i = 1; i <= oMatrixCliente.RowCount; i++)
    {
        if (oMatrixCliente.IsRowSelected(i))
            break;
    }

    SAPbobsCOM.Recordset RS;

    RS = oCompany.GetBusinessObject(SAPbobsCOM.BoObjectTypes.BoRecordset);
    if (posicao == 0)
        sSQL = "SELECT U_numero, U_grupo FROM dbo.[@TESEGC] WHERE U_numero = " +
        ((SAPbouiCOM.EditText)oMatrixCliente.Columns.Item("V_1").Cells.Item(i).Specific).Value.ToString() + """;
    else
        sSQL = "SELECT U_numero, U_grupo FROM dbo.[@TESEGC] WHERE U_numero = " + posicao + """;

    RS.DoQuery(sSQL);
    if (RS.RecordCount > 0)
    {
        RS.MoveFirst();

        sSQL = System.Convert.ToString(RS.Fields.Item("U_numero").Value);
        SBO_Application.MessageBox(sSQL, 1, "OK", "", "");
        UTS = oCompany.UserTables;
        UT = UTS.Item("TESEGC");
        UT.GetByKey(sSQL);
        UT.Remove();
    }
    oMatrixCliente.DeleteRow(i);
}

private void RemoverGclienteTotal(int lol)

```



```

{
    SAPbobsCOM.Recordset RS;

    RS = oCompany.GetBusinessObject(SAPbobsCOM.BoObjectTypes.BoRecordset);

    sSQL = "SELECT * FROM dbo.[@TESEGC] WHERE U_campanha = " + lol + """;

    RS.DoQuery(sSQL);
    RS.MoveFirst();
    while (RS.EoF == false)
    {
        sSQL = System.Convert.ToString(RS.Fields.Item("U_numero").Value);
        UTS = oCompany.UserTables;
        UT = UTS.Item("TESEGC");
        UT.GetByKey(sSQL);
        UT.Remove();
        RS.MoveNext();
    }
}

private void RemoverGartigo(int posicao)
{
    int i;
    for (i = 1; i <= oMatrixArtigo.RowCount; i++)
    {
        if (oMatrixArtigo.IsRowSelected(i))
            break;
    }

    SAPbobsCOM.Recordset RS;

    RS = oCompany.GetBusinessObject(SAPbobsCOM.BoObjectTypes.BoRecordset);

    sSQL = "SELECT U_numero, U_grupo FROM dbo.[@TESEGA] WHERE U_numero = " +
    ((SAPbouiCOM.EditText)oMatrixArtigo.Columns.Item("V_1").Cells.Item(i).Specific).Value.ToString() + """;

    RS.DoQuery(sSQL);
    if (RS.RecordCount > 0)
    {
        RS.MoveFirst();

        sSQL = System.Convert.ToString(RS.Fields.Item("U_numero").Value);
        SBO_Application.MessageBox(sSQL, 1, "OK", "", "");
        UTS = oCompany.UserTables;
        UT = UTS.Item("TESEGA");
        UT.GetByKey(sSQL);
        UT.Remove();
    }
    //if(posicao == 0)
    if(oButton2.Caption == "Adicionar")

    RemoveTabelaExcepcoes(((SAPbouiCOM.EditText)oMatrixArtigo.Columns.Item("V_0").Cells.Item(i).Specific).Value.ToString(),
    ((SAPbouiCOM.EditText)oMatrixArtigo.Columns.Item("V_2").Cells.Item(i).Specific).Value.ToString());
    else if (oButton2.Caption == "Gravar")

    RemoveTabelaExcepcoes(((SAPbouiCOM.EditText)oMatrixArtigo.Columns.Item("V_0").Cells.Item(i).Specific).Value.ToString(), editcampanha.ToString());
    //RemoverTabelaExcepcoesTotal(posicao.ToString());

    //if(posicao == 0)
    oMatrixArtigo.DeleteRow(i);
}

```

```

private void RemoverGartigoTotal(int lol)
{
    SAPbobsCOM.Recordset RS;

    RS = oCompany.GetBusinessObject(SAPbobsCOM.BoObjectTypes.BoRecordset);

    sSQL = "SELECT * FROM dbo.[@TESEGA] WHERE U_campanha = " + lol + """;

    RS.DoQuery(sSQL);
    RS.MoveFirst();
    while (RS.EoF == false)
    {
        sSQL = System.Convert.ToString(RS.Fields.Item("U_numero").Value);
        UTS = oCompany.UserTables;
        UT = UTS.Item("TESEGA");
        UT.GetByKey(sSQL);
        UT.Remove();
        RS.MoveNext();
    }
}

private int RemoverSelecionada()
{
    for (int i = 1; i <= oMatrix.RowCount; i++)
    {
        if (oMatrix.IsRowSelected(i))
            return i;
    }
    return 0;
}

private void RemoverCampanha()
{
    int j, a;
    oRecordset = ( ( SAPbobsCOM.Recordset ) ( oCompany.GetBusinessObject(
SAPbobsCOM.BoObjectTypes.BoRecordset ) ) );
    j = RemoverSelecionada();
    sSQL = "SELECT U_Numero, U_Estado FROM dbo.[@TESE] WHERE U_Numero = " +
((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_0").Cells.Item(j).Specific).Value.ToString()+"";
    a =
Convert.ToInt32(((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_0").Cells.Item(j).Specific).Value.ToString());

    oRecordset.DoQuery( sSQL );
    // Check if a record exists with this [ItemCode] and [SuppSerial]
    if (oRecordset.RecordCount > 0)
    {
        oRecordset.MoveFirst();

        sSQL = System.Convert.ToString(oRecordset.Fields.Item("U_Numero").Value);

        UTS = oCompany.UserTables;
        UT = UTS.Item("TESE");
        UT.GetByKey(sSQL);
        UT.Remove();
        SBO_Application.MessageBox("Campanha Removida", 1, "OK", "", "");
    }
    //RemoverGartigo(a);
    RemoverGclienteTotal(a);
    RemoverGartigoTotal(a);
    RemoverTabelaExcepcoesTotal(a);
    oMatrix.DeleteRow(j);
}

```

```

#endregion

#region PublicarEstado

public void LoadVariveis()
{
    oForm = SBO_Application.Forms.Item("frmSegundo");
    // Create a user Datasource for the checkboxes
    userDB = oForm.DataSources.UserDataSources.Add("oCheckDS",
SAPbouiCOM.BoDataType.dt_SHORT_TEXT, 1);
    userDB1 = oForm.DataSources.UserDataSources.Add("oCheckDS1",
SAPbouiCOM.BoDataType.dt_SHORT_TEXT, 1);
    userDB2 = oForm.DataSources.UserDataSources.Add("oCheckDS2",
SAPbouiCOM.BoDataType.dt_SHORT_TEXT, 1);
    userDB3 = oForm.DataSources.UserDataSources.Add("oCheckDS3",
SAPbouiCOM.BoDataType.dt_SHORT_TEXT, 1);
    userDB4 = oForm.DataSources.UserDataSources.Add("oCheckDS4",
SAPbouiCOM.BoDataType.dt_SHORT_TEXT, 1);

    // Create a user Datasource for the Option button
    formDB = oForm.DataSources.UserDataSources.Add("optnDS",
SAPbouiCOM.BoDataType.dt_SHORT_TEXT, 1);
    // Create DBDataSource
    oDBDataSource = oForm.DataSources.DBDataSources.Add("OCRD");
    oDBSource = oForm.DataSources.DBDataSources.Add("OITM");

    //Option Button
    oItem = oForm.Items.Item("checkinit1");
    Opt1 = ((SAPbouiCOM.OptionBtn)(oItem.Specific));
    Opt1.GroupWith("checkfim1");
    Opt1.DataBind.SetBound(true, "", "optnDS");

    oItem = oForm.Items.Item("checkfim1");
    Optt2 = ((SAPbouiCOM.OptionBtn)(oItem.Specific));
    Optt2.GroupWith("checkinit1");
    Optt2.DataBind.SetBound(true, "", "optnDS");

    oItem = oForm.Items.Item("ComboBox2");
    oCombo1 = ((SAPbouiCOM.ComboBox)(oItem.Specific));

    //EditBox do Texto Extra
    oItem = oForm.Items.Item("ExtEdit1");
    ExtEdit = oItem.Specific;

    //Botão do Load
    oItem = oForm.Items.Item("Load1");
    oButton1 = oItem.Specific;

    //Botao do Limpar Seleção
    oItem = oForm.Items.Item("Limpar1");
    oButton2 = oItem.Specific;

    //Botao do Selecionar todos
    oItem = oForm.Items.Item("Sele1");
    oButton3 = oItem.Specific;

    //Botao do Upload Imagem
    oItem = oForm.Items.Item("browse1");
    oButton4 = oItem.Specific;

    //Botao do Publicar
    oItem = oForm.Items.Item("Button1");
    oButton5 = oItem.Specific;

    //Botao do Sair
    oItem = oForm.Items.Item("Sair");
}

```

```

oButton6 = oItem.Specific;

//EditBox da foto
oItem = oForm.Items.Item("imagelink1");
oLink = ((SAPbouiCOM.EditText)(oItem.Specific));

//EditBox da Pesquisa
oItem = oForm.Items.Item("EditBox1");
oEdit = ((SAPbouiCOM.EditText)(oItem.Specific));

//Static Box da pesquisa
oItem = oForm.Items.Item("5");
StaText1 = ((SAPbouiCOM.StaticText)(oItem.Specific));

//Static Box do Acrescentar Texto
oItem = oForm.Items.Item("19");
StaText2 = ((SAPbouiCOM.StaticText)(oItem.Specific));

//Static do link da imagem
oItem = oForm.Items.Item("21");
StaText3 = ((SAPbouiCOM.StaticText)(oItem.Specific));

//Static Box dos campos
oItem = oForm.Items.Item("campos");
StaText = ((SAPbouiCOM.StaticText)(oItem.Specific));

//ComboBox do Load
oItem = oForm.Items.Item("ComboBox1");
oCombo = ((SAPbouiCOM.ComboBox)(oItem.Specific));
PreencherComboBoxLoad();

//CheckBoxs
oItem = oForm.Items.Item("checktodo1");
oCheck = ((SAPbouiCOM.CheckBox)(oItem.Specific));
oCheck.ValOn = "Y";
oCheck.DataBind.SetBound(true, "", "oCheckDS");

oItem = oForm.Items.Item("checkcol1");
oCheck1 = ((SAPbouiCOM.CheckBox)(oItem.Specific));
oCheck1.ValOn = "Y";
oCheck1.DataBind.SetBound(true, "", "oCheckDS1");

oItem = oForm.Items.Item("checkcol2");
oCheck2 = ((SAPbouiCOM.CheckBox)(oItem.Specific));
oCheck2.ValOn = "Y";
oCheck2.DataBind.SetBound(true, "", "oCheckDS2");

oItem = oForm.Items.Item("checkcol3");
oCheck3 = ((SAPbouiCOM.CheckBox)(oItem.Specific));
oCheck3.ValOn = "Y";
oCheck3.DataBind.SetBound(true, "", "oCheckDS3");

oItem = oForm.Items.Item("checkcol4");
oCheck4 = ((SAPbouiCOM.CheckBox)(oItem.Specific));
oCheck4.ValOn = "Y";
oCheck4.DataBind.SetBound(true, "", "oCheckDS4");

oCheck.Checked = false;
oCheck1.Checked = false;
oCheck2.Checked = false;
oCheck3.Checked = false;
oCheck4.Checked = false;
}

public void PreencherMatrixClientes()
{
    LimparMatrix();
}

```

```

checkboxCliente();

oDBDataSource = oForm.DataSources.DBDataSources.Add("OCRD");
oItem = oForm.Items.Item("mat");
oMatrix = ((SAPbouiCOM.Matrix)(oItem.Specific));
oColumns = oMatrix.Columns;

oColumn1 = oColumns.Item("V_3");
oColumn1.TitleObject.Caption = "Nome Cliente";
oColumn1.DataBind.SetBound(true, "OCRD", "CardName");

oColumn2 = oColumns.Item("V_2");
oColumn2.TitleObject.Caption = "Tipo de Parceiro";
oColumn2.DataBind.SetBound(true, "OCRD", "CardType");

oColumn3 = oColumns.Item("V_1");
oColumn3.TitleObject.Caption = "Web Site";
oColumn3.DataBind.SetBound(true, "OCRD", "IntrntSite");

oColumn4 = oColumns.Item("V_0");
oColumn4.TitleObject.Caption = "Morada";
oColumn4.DataBind.SetBound(true, "OCRD", "Address");

oDBDataSource.Query(null);
oMatrix.LoadFromDataSource();

for (int i = 1; i <= oMatrix.RowCount; i++)
{
    if (((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_2").Cells.Item(i).Specific).Value.ToString() !=
"C")
    {
        oMatrix.DeleteRow(i);
        i--;
    }
}

private void PreencherMatrixArtigos()
{
    LimparMatrix();
    checkboxArtigos();

    oItem = oForm.Items.Item("mat");
    oMatrix = ((SAPbouiCOM.Matrix)(oItem.Specific));
    oColumns = oMatrix.Columns;

    oColumn1 = oColumns.Item("V_3");
    oColumn1.TitleObject.Caption = "Nome Artigo";
    oColumn1.DataBind.SetBound(true, "OITM", "ItemName");

    oColumn2 = oColumns.Item("V_2");
    oColumn2.TitleObject.Caption = "Stock";
    oColumn2.DataBind.SetBound(true, "OITM", "OnHand");

    oColumn3 = oColumns.Item("V_1");
    oColumn3.TitleObject.Caption = "Preço";
    oColumn3.DataBind.SetBound(true, "OITM", "LastPurPrc");

    oColumn4 = oColumns.Item("V_0");
    oColumn4.TitleObject.Caption = "Código IVA";
    oColumn4.DataBind.SetBound(true, "OITM", "VatGourpSa");

    oDBDSource.Query(null);
    oMatrix.LoadFromDataSource();
}

```

```

private void PreencherMatrixFornecedores()
{
    LimparMatrix();
    checkboxCliente();
    oItem = oForm.Items.Item("mat");
    oMatrix = ((SAPbouiCOM.Matrix)oItem.Specific);
    oColumns = oMatrix.Columns;

    oColumn1 = oColumns.Item("V_3");
    oColumn1.TitleObject.Caption = "Nome Fornecedores";
    oColumn1.DataBind.SetBound(true, "OCRD", "CardName");

    oColumn2 = oColumns.Item("V_2");
    oColumn2.TitleObject.Caption = "Tipo de Parceiro";
    oColumn2.DataBind.SetBound(true, "OCRD", "CardType");

    oColumn3 = oColumns.Item("V_1");
    oColumn3.TitleObject.Caption = "Web Site";
    oColumn3.DataBind.SetBound(true, "OCRD", "IntrntSite");

    oColumn4 = oColumns.Item("V_0");
    oColumn4.TitleObject.Caption = "Morada";
    oColumn4.DataBind.SetBound(true, "OCRD", "Address");

    oDBDataSource.Query(null);
    oMatrix.LoadFromDataSource();

    for (int i = 1; i <= oMatrix.RowCount; i++)
    {
        if (((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_2").Cells.Item(i).Specific).Value.ToString() !=
"S")
        {
            oMatrix.DeleteRow(i);
            i--;
        }
    }
}

private void PreencherMatrixLead()
{
    LimparMatrix();
    checkboxCliente();
    oItem = oForm.Items.Item("mat");
    oMatrix = ((SAPbouiCOM.Matrix)oItem.Specific);
    oColumns = oMatrix.Columns;

    oColumn1 = oColumns.Item("V_3");
    oColumn1.TitleObject.Caption = "Nome Lead";
    oColumn1.DataBind.SetBound(true, "OCRD", "CardName");

    oColumn2 = oColumns.Item("V_2");
    oColumn2.TitleObject.Caption = "Tipo de Parceiro";
    oColumn2.DataBind.SetBound(true, "OCRD", "CardType");

    oColumn3 = oColumns.Item("V_1");
    oColumn3.TitleObject.Caption = "Web Site";
    oColumn3.DataBind.SetBound(true, "OCRD", "IntrntSite");

    oColumn4 = oColumns.Item("V_0");
    oColumn4.TitleObject.Caption = "Morada";
    oColumn4.DataBind.SetBound(true, "OCRD", "Address");

    oDBDataSource.Query(null);
    oMatrix.LoadFromDataSource();

    for (int i = 1; i <= oMatrix.RowCount; i++)

```

```

        {
            if (((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_2").Cells.Item(i).Specific).Value.ToString() !=
"L")
            {
                oMatrix.DeleteRow(i);
                i--;
            }
        }
    }

private void PreencherComboBoxLoad()
{
    oCombo.ValidValues.Add("Clientes", "Lista de Clientes");
    oCombo.ValidValues.Add("Fornecedores", "Lista de Fornecedores");
    oCombo.ValidValues.Add("Lead", "Lista de Leads");
    oCombo.ValidValues.Add("Artigos", "Lista de Artigos");
}

private void PreencherComboBoxPesquisa()
{
    if (pesquisa == 0)
    {
        if (artigos == 1)
        {
            RemoverPesquisaArtigos();
        }

        oCombo1.ValidValues.Add("Nome", "Nome");
        oCombo1.ValidValues.Add("Tipo", "Tipo");
        oCombo1.ValidValues.Add("WebSite", "WebSite");
        oCombo1.ValidValues.Add("Morada", "Morada");
        pesquisa = 1;
        artigos = 0;
    }
}

private void PreencherComboBoxPesquisaArtigos()
{
    oItem = oForm.Items.Item("ComboBox2");
    oCombo1 = ((SAPbouiCOM.ComboBox)(oItem.Specific));

    if (artigos == 0)
    {
        if (pesquisa == 1)
        {
            RemoverPesquisaClientes();
        }
        oCombo1.ValidValues.Add("Nome", "Nome");
        oCombo1.ValidValues.Add("Stock", "Stock");
        oCombo1.ValidValues.Add("Preço", "Preço");
        oCombo1.ValidValues.Add("Código IVA", "Código IVA");
        artigos = 1;
        pesquisa = 0;
    }
}

private void PesquisaClientes(String pesquisa, String tipo)
{
    oMatrix.ClearSelections();
    if (tipo == "Nome")
    {
        for (int i = 1; i <= oMatrix.RowCount; i++)
        {
            if (((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_3").Cells.Item(i).Specific).Value.ToString() ==
pesquisa)
            {

```

```

        oMatrix.SelectRow(i, true, true);
    }
}

if (tipo == "Tipo")
{
    for (int i = 1; i <= oMatrix.RowCount; i++)
    {
        if (((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_2").Cells.Item(i).Specific).Value.ToString() ==
pesquisa)
        {
            oMatrix.SelectRow(i, true, true);
        }
    }
}

if (tipo == "WebSite")
{
    for (int i = 1; i <= oMatrix.RowCount; i++)
    {
        if (((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_1").Cells.Item(i).Specific).Value.ToString() ==
pesquisa)
        {
            oMatrix.SelectRow(i, true, true);
        }
    }
}

if (tipo == "Morada")
{
    for (int i = 1; i <= oMatrix.RowCount; i++)
    {
        if (((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_0").Cells.Item(i).Specific).Value.ToString() ==
pesquisa)
        {
            oMatrix.SelectRow(i, true, true);
        }
    }
}

if (tipo == "Stock")
{
    for (int i = 1; i <= oMatrix.RowCount; i++)
    {
        if (((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_2").Cells.Item(i).Specific).Value.ToString() ==
pesquisa)
        {
            oMatrix.SelectRow(i, true, true);
        }
    }
}

if (tipo == "Preço")
{
    for (int i = 1; i <= oMatrix.RowCount; i++)
    {
        if (((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_1").Cells.Item(i).Specific).Value.ToString() ==
pesquisa)
        {
            oMatrix.SelectRow(i, true, true);
        }
    }
}

if (tipo == "Código IVA")
{
    for (int i = 1; i <= oMatrix.RowCount; i++)
    {

```



```

        if (((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_0").Cells.Item(i).Specific).Value.ToString() ==
pesquisa)
        {
            oMatrix.SelectRow(i, true, true);
        }
    }
}

private int LinhaSelecionada()
{
    for (int i = 1; i <= oMatrix.RowCount; i++)
    {
        if (oMatrix.IsRowSelected(i))
            return i;
    }

    return 0;
}

private void Compilar(int check1, int check2, int check3, int check4, int check5)
{
    int i;
    int j = 0;
    int h = 0;
    i = LinhaSelecionada();

    if (Opt1.Selected == true)
    {
        temp2 = ExtEdit.String;
        temp2 += "\n \n";
    }

    if (check1 == 2)
    {
        temp = ((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_3").Cells.Item(i).Specific).Value.ToString();
        temp += "\n \n";
        temp += ((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_2").Cells.Item(i).Specific).Value.ToString();
        temp += "\n \n";
        temp += ((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_1").Cells.Item(i).Specific).Value.ToString();
        temp += "\n \n";
        temp += ((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_0").Cells.Item(i).Specific).Value.ToString();
    }
    else
    {
        if (check2 == 2)
        {
            temp
            ((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_3").Cells.Item(i).Specific).Value.ToString();
            j++;
            h = 2;
        }
        if (check3 == 2 && j == 0)
        {
            temp
            ((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_2").Cells.Item(i).Specific).Value.ToString();
            j++;
            h = 3;
        }
        if (check4 == 2 && j == 0)
        {
            temp
            ((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_1").Cells.Item(i).Specific).Value.ToString();
            j++;
            h = 4;
        }
    }
}

```

```

        if (check5 == 2 && j == 0)
        {
            temp
            ((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_0").Cells.Item(i).Specific).Value.ToString();
            j++;
            h = 5;
        }
        if (h == 2)
        {
            if (check3 == 2)
            {
                temp += "\n \n";
                temp
                ((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_2").Cells.Item(i).Specific).Value.ToString();
            }

            if (check4 == 2)
            {
                temp += "\n \n";
                temp
                ((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_1").Cells.Item(i).Specific).Value.ToString();
            }

            if (check5 == 2)
            {
                temp += "\n \n";
                temp
                ((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_0").Cells.Item(i).Specific).Value.ToString();
            }
        }

        if (h == 3)
        {
            if (check4 == 2)
            {
                temp += "\n \n";
                temp
                ((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_1").Cells.Item(i).Specific).Value.ToString();
            }

            if (check5 == 2)
            {
                temp += "\n \n";
                temp
                ((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_0").Cells.Item(i).Specific).Value.ToString();
            }
        }

        if (h == 4)
        {
            if (check5 == 2)
            {
                temp += "\n \n";
                temp
                ((SAPbouiCOM.EditText)oMatrix.Columns.Item("V_0").Cells.Item(i).Specific).Value.ToString();
            }
        }

        if (Optt2.Selected == true)
        {
            temp3 = ExtEdit.String;
        }
        temp2 += temp;
        temp2 += "\n \n";
        temp2 += temp3;
        Publicar();

```

```

}

private void Publicar()
{
    try
    {
        FacebookAPI api = new Facebook.FacebookAPI(myToken.Default.token);
        JSONObject me = api.Get("/me");
        var yo = me.Dictionary["id"].String;
        myToken.Default.userId = yo.ToString();
        Dictionary<string, string> postArgs = new Dictionary<string, string>();
        postArgs["message"] = temp2;
        JSONObject post = api.Post("/" + myToken.Default.userId + "/feed", postArgs);
        oEdit.String = "";
        SBO_Application.MessageBox("Enviado", 1, "OK", "", "");
        temp = "";
        temp2 = "";
        temp3 = "";

    }
    catch (FacebookAPIException exr)
    {
        Console.WriteLine(exr.Message);
    }
}

private void RemoverPesquisaClientes()
{
    oCombo1.ValidValues.Remove("Nome");
    oCombo1.ValidValues.Remove("Tipo");
    oCombo1.ValidValues.Remove("WebSite");
    oCombo1.ValidValues.Remove("Morada");
}

private void RemoverPesquisaArtigos()
{
    oCombo1.ValidValues.Remove("Nome");
    oCombo1.ValidValues.Remove("Stock");
    oCombo1.ValidValues.Remove("Preço");
    oCombo1.ValidValues.Remove("Código IVA");
}

private void LimparMatrix()
{
    oItem = oForm.Items.Item("mat");
    oMatrix = ((SAPbouiCOM.Matrix)(oItem.Specific));
    oMatrix.Clear();
    StaText.Item.Visible = true;
}

private void checkboxCliente()
{
    oCheck.Caption = "Todos";
    oCheck.Item.Visible = true;
    oCheck1.Caption = "Nome";
    oCheck1.Item.Visible = true;
    oCheck2.Caption = "Tipo";
    oCheck2.Item.Visible = true;
    oCheck3.Caption = "Web Site";
    oCheck3.Item.Visible = true;
    oCheck4.Caption = "Morada";
    oCheck4.Item.Visible = true;
}

private void checkboxArtigos()
{
    oCheck.Caption = "Todos";

```

```
oCheck.Item.Visible = true;
oCheck1.Caption = "Nome";
oCheck1.Item.Visible = true;
oCheck2.Caption = "Stock";
oCheck2.Item.Visible = true;
oCheck3.Caption = "Preço";
oCheck3.Item.Visible = true;
oCheck4.Caption = "IVA";
oCheck4.Item.Visible = true;
}
#endregion
}
}
```