

**FACULDADES INTEGRADAS DE TAQUARA
CURSO DE SISTEMAS DE INFORMAÇÃO**

SISCORE: UMA APLICAÇÃO WEB PARA *BALANCED SCORECARD*

LEANDRO SORGETZ

Taquara

2009

LEANDRO SORGETZ

SISCORE: UMA APLICAÇÃO WEB PARA *BALANCED SCORECARD*

Trabalho de Conclusão apresentado ao Curso de Sistemas de Informação das Faculdades Integradas de Taquara, como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação, sob orientação do Prof. Mestre Francisco Assis Moreira do Nascimento.

Taquara

2009

Dedico este trabalho à Rosângela, minha querida e amada esposa, que corajosamente esteve ao meu lado tanto nos melhores momentos da minha vida como nos mais difíceis. E jamais me abandonou.

AGRADECIMENTOS

Agradeço a meu Pai Celeste, autor da vida, que me sustentou e ainda me sustenta durante todos os dias da minha vida; a minha esposa Rosângela, pela paciência e apoio irrestrito; aos meus filhos Laize, Rebecca e Norton pelas alegrias que eles me dão e pelo prazer em viver que eles transmitem e mesmo pequeninos e sentindo minha ausência, percebi sua compreensão. Aos meus queridos pais, Selito e Celita e a meu irmão Fernando, por sempre me apoiarem e acreditarem no meu potencial. Agradeço aos meus queridos e amados amigos Édison e Carla por terem me dedicado tanto tempo de suas vidas.

Ao Professor Delmar Henrique Backes, por ter me dado crédito e proporcionado uma oportunidade quando muitos não o fariam. A todos os professores que contribuíram para que esse trabalho se realizasse, especialmente ao meu colega e orientador Professor Francisco Assis Moreira do Nascimento que nunca mediu esforços para que esse e outros trabalhos se realizassem. Para os meus colegas de trabalho pela parceria e amizade, especialmente para minha querida amiga e colega Professora Marli Schuller Castro pelo carinho e incentivo.

E aos meus colegas e professores de curso, especialmente aos que estão concluindo essa trajetória comigo: Naira, Luciano, Ramão, Nice, Alencar e Israel. Valeu galerinha!

RESUMO

Este trabalho tem a finalidade de descrever o desenvolvimento do SISCORE, uma ferramenta Web direcionada a implementação do *Balanced Scorecard* – uma metodologia para aprimoramento do planejamento estratégico. O uso dessa metodologia se expandiu desde a sua criação e tem auxiliado diversos segmentos do meio empresarial a evoluírem frente aos desafios e exigências de um mercado cada dia mais competitivo. Os casos de sucesso motivam mais empresas aderirem a essa metodologia e estimulam a indústria do *software* a desenvolver ferramentas cada vez mais inteligentes, complexas e, conseqüentemente, mais caras. O alto custo é um fator que diminui a adesão de pequenas e médias empresas, considerando que isso, cada vez mais, precisa ser feito através de uma ferramenta específica para tal. Pode-se encontrar opções de código aberto para diversos *softwares*, inclusive para *Balanced Scorecard*, entretanto, poucas dessas ferramentas contemplam essa metodologia, ou não possuem documentação suficiente para colocá-las em pleno funcionamento. O SISCORE é um *software* direcionado para pequenas e médias empresas por apresentar um custo de desenvolvimento acessível. O baixo custo é atribuído ao uso de ferramentas livres e de código aberto. O desenvolvimento do SISCORE iniciou com a escolha das tecnologias a serem usadas. Nesse estágio foi definida PHP como linguagem de programação. Também foi escolhido o *framework* Yii para a aplicação do padrão de projeto *Model-View-Controller* (Modelo-Controle-Visão), todos usados em um ambiente integrado de desenvolvimento. O levantamento e análise de requisitos, necessários para implementar um aplicativo, usou conceitos do Processo Unificado. A documentação foi feita usando dos elementos e diagramas de UML - *Unified Modeling Language* (Linguagem de Modelagem Unificada). Para os testes do SISCORE foi aplicado o tipo de testes denominado “caixa-preta”. Para obter os resultados do uso do SISCORE, foi seguindo o padrão fornecido pelo INEP – Instituto Nacional de Estudos e Pesquisas Educacionais.

Palavras-chave: *Balanced Scorecard*. Planejamento Estratégico, Ferramenta Web, Código Aberto, Ferramentas Livres, Processo Unificado, *Framework*, *Model-View-Controller*, Padrões de Projeto.

SUMÁRIO

1	INTRODUÇÃO.....	9
2	FUNDAMENTOS.....	12
2.1	Balanced Scorecard.....	12
2.1.1	Histórico.....	12
2.1.2	Conceitos.....	14
2.1.2.1	<i>Estratégia.....</i>	<i>14</i>
2.1.2.2	<i>Mapa Estratégico.....</i>	<i>14</i>
2.1.2.3	<i>Medidas de desempenho.....</i>	<i>16</i>
2.1.2.4	<i>Metas.....</i>	<i>17</i>
2.1.2.5	<i>Plano de Ação.....</i>	<i>18</i>
2.1.2.6	<i>O Balanced Scorecard na estruturação do planejamento estratégico.....</i>	<i>18</i>
2.2	Metodologias para desenvolvimento.....	20
2.2.1	Padrão de projeto MVC.....	20
2.2.2	Ajax.....	21
2.2.3	Orientação a objetos.....	22
2.2.4	Processo de software.....	23
2.2.4.1	<i>Modelo sequencial linear.....</i>	<i>23</i>
2.2.4.2	<i>Modelo de prototipagem.....</i>	<i>24</i>
2.2.4.3	<i>Modelo RAD.....</i>	<i>24</i>
2.2.4.4	<i>Modelo de software evolucionário.....</i>	<i>25</i>
2.3	Tecnologias para Desenvolvimento de Aplicações Web.....	25
2.3.1	Linguagens.....	26
2.3.1.1	<i>PHP.....</i>	<i>26</i>
2.3.1.2	<i>HTML.....</i>	<i>27</i>
2.3.1.3	<i>CSS.....</i>	<i>27</i>
2.3.1.4	<i>XML.....</i>	<i>28</i>

2.3.1.5	<i>Javascript</i>	28
2.3.1.6	<i>Banco de dados</i>	28
2.3.1.7	<i>UML</i>	29
2.3.2	<i>Frameworks</i>	30
2.3.2.1	<i>Comparação entre Frameworks</i>	31
2.3.2.2	<i>Conclusões das Comparações entre Frameworks</i>	35
3	TRABALHOS CORRELATOS	36
3.1	Ferramentas de BSC Específicas	36
3.1.1	BSPG.....	37
3.1.2	OpenScorecard	38
3.1.3	BlueOxygen.....	39
3.1.4	Strategic Adviser.....	39
3.2	Softwares BSC Embutidos.....	41
3.2.1	SAP	41
3.2.2	CIGAM.....	42
3.2.3	OpenBravo.....	44
3.2.4	Adempiere.....	44
3.3	Comparativo das Ferramentas de BSC Existentes.....	46
3.4	Casos de implementação do BSC.....	48
3.4.1	O processo na Werner Calçados.....	48
3.5	O início do processo.....	49
3.5.1.1	<i>Segundo passo - Conscientização</i>	49
3.5.1.2	<i>Desenvolvimento</i>	49
3.5.1.3	<i>A prática</i>	50
3.5.1.4	<i>Ferramentas</i>	50
3.5.1.5	<i>Conclusões do caso Werner Calçados</i>	51
3.5.2	O caso Grendene.....	51
3.5.2.1	<i>O desafio</i>	51
3.5.2.2	<i>Ferramentas</i>	52
3.5.2.3	<i>Os resultados do caso Grendene</i>	52

4	SISCORE: UMA FERRAMENTA WEB PARA BSC.....	53
4.1	Descrição do SISCORE.....	53
4.2	Desenvolvimento do SISCORE.....	54
4.2.1	Análise.....	54
4.2.1.1	<i>Requisitos Funcionais.....</i>	<i>55</i>
4.2.1.2	<i>Requisitos Não Funcionais.....</i>	<i>56</i>
4.2.2	Projeto.....	56
4.2.2.1	<i>Casos de Uso.....</i>	<i>57</i>
4.2.2.3	<i>Diagramas de Classe.....</i>	<i>61</i>
4.2.2.4	<i>Projeto E-R.....</i>	<i>65</i>
4.2.3	Codificação	67
4.2.3.1	<i>Estrutura da aplicação.....</i>	<i>68</i>
4.2.3.2	<i>Estrutura de diretórios.....</i>	<i>70</i>
4.2.4	Testes	71
5	UTILIZAÇÃO DO SISCORE.....	72
5.1	Padrões de avaliação SINAES.....	72
5.2	Perspectivas e indicadores.....	72
5.3	Tela de acesso.....	74
5.4	Tela inicial	75
5.5	Gerenciamento dos cadastros.....	76
5.6	Criação de um BSC com o SISCORE.....	78
5.7	Monitoramento dos resultados.....	80
5.8	Visualizando resultados através de gráficos.....	81
6	CONCLUSÕES.....	84
7	REFERÊNCIAS.....	86
	APÊNDICES.....	89
	APÊNDICE A – Detalhamento de um caso de uso.....	90

1 INTRODUÇÃO

O meio empresarial enfrenta muitos desafios por conta do aumento da competitividade. Fatores como esse estimulam práticas de planejamento estratégico cada vez mais refinadas. Existem diversas ferramentas e metodologias que buscam contemplar essas práticas, mas Müller (2001) já conclui que o uso isolado das ferramentas e resultados provenientes da contabilidade tradicional, que não consideram a individualidade da empresa, pouco contribuem para seu sucesso. É necessário desenvolver metodologias direcionadas ao momento e às necessidades da empresa.

O uso de planos de ação e estratégias condizentes com os objetivos e metas de uma empresa são iniciativas favoráveis às necessidades de posicionar-se ostensivamente frente às exigências mercadológicas atuais. Dentre as principais ferramentas de gestão estratégica encontra-se o *Balanced Scorecard* – BSC que segundo Kaplan e Norton (1997, p. 24) é uma “[...] ferramenta completa que traduz a visão e a estratégia da empresa num conjunto coerente de medidas de desempenho”. É um importante instrumento para ambientes gerenciais que buscam aprimorar suas estratégias a partir de medidas de desempenho com ênfase nos resultados.

A implementação de um BSC, através de uma ferramenta computacional específica, auxilia e automatiza o acompanhamento e apresentação dos resultados através de relatórios e gráficos descritivos do andamento da implantação e da efetivação dos objetivos. Isso agrega valor ao método tornando os resultados mais assertivos e sem a morosidade de métodos tradicionais como o uso de planilhas ocasionam.

Existem iniciativas que vislumbram tal tarefa, entretanto estão muito direcionadas a negócios mais específicos ou estão integradas em aplicações ERP¹ e em outros aplicativos de gestão empresarial. Um ERP, como o da SAP², tem um custo muito alto e muitas empresas não têm condições ou mesmo não estão dispostas a investir quantias tão elevadas em ferramentas como essa. Outras *softwares* como o ERP da CIGAM³ tem custos mais acessíveis, mas apresentam apenas algumas características de ERP.

A relação custo/benefício é um fator importante. Um dos objetivos deste projeto é

¹ *Software* integrado que abrange todos os processos de uma empresa.

² *Systems Applications and Products* - Empresa alemã de soluções empresariais (<http://www.sap.com>).

³ Empresa brasileira de soluções para gestão empresarial (<http://www.cigam.com.br>).

oportunizar a implementação de um BSC em empresas de pequeno e médio porte. A grande maioria destas empresas não possui recursos para aplicar em sistemas complexos como um ERP da SAP, mesmo que ele já incorpore versões chamadas “*standard*” ou modularizadas que são mais acessíveis; ressaltando que essas versões têm também os recursos reduzidos.

Em muitos casos a intenção não é adquirir uma ferramenta para ERP que atenda à demanda por um BSC, seja ele eletrônico ou não, mas incorporar um aplicativo ao próprio legado do sistema de informações empresarial. Um caso de ferramenta desenvolvida especificamente para a implantação de um BSC é a ferramenta *Strategic Adviser – SA*⁴, mas não é desenvolvida para Web.

O BSC é uma metodologia de gestão que consolidou seu uso em muitas empresas e já se encontra em um estágio bastante evoluído. A forma como esse amadurecimento ocorreu tornou-o tradicional e já requer certas mudanças em seu escopo.

A proposta inicial do BSC trabalha com quatro perspectivas: (i) financeira, (ii) cliente, (iii) processos internos e (iv) aprendizado e crescimento. Kaplan e Norton (1997) criaram o BSC em 1992 e seu uso, no passar dos anos, leva diferentes visões de negócio à possibilidade de acrescentar novas perspectivas como, por exemplo, a social e a do meio ambiente.

A aplicação do BSC pode tornar-se bastante ampla como também pontual. Um exemplo como esse ocorreu na Grendene S.A., onde o CIO Ernani Toso conheceu, estudou, capacitou sua equipe e implementou o BSC na área de TI desta empresa. Este caso é relatado no Seção 3.

A necessidade de disponibilizar um *software* usando tecnologias livres e de código aberto para BSC e que ofereça acesso remoto é um dos motivos que impulsionam esse trabalho, uma vez que não foram encontradas iniciativas que vislumbrem tal proposta. A inclusão de novas perspectivas bem como as abordagens que surgem do seu uso, estão previstas na implementação do SISCORE, tornando este mais um aspecto importante no desenvolvimento de ferramentas para esse fim, assim como o fato de ser um sistema desenvolvido para Web e de *Open Source* (código aberto).

A eficiência e o aumento do uso do BSC, desde sua criação, apontam para a necessidade de estimular mais empresas a implementarem essa ferramenta estratégica por meio de um sistema como o SISCORE, assim como facilitar o trabalho daqueles que já o

⁴ *Interact Solutions* – Empresa brasileira de soluções estratégicas integradas (<http://www.interact.com.br>).

fizeram, mas com outras formas que não por uma aplicação específica para esse fim, por exemplo, através planilhas de cálculo.

O SISCORE é uma aplicação que propõe facilitar a implantação do BSC focando no auxílio da gestão de seus objetivos, indicadores, metas, iniciativas e o próprio planejamento estratégico. A proposta é não ser apenas um registrador de indicadores, com a apresentação de alguns mapas e gráficos ou que faça comparações óbvias entre eles, mas sim focar na gestão destes indicadores, cruzando informações para destacar e sinalizar a evolução das metas e objetivos propostos.

A próxima Seção apresenta os principais conceitos que norteiam o BSC, tais como indicadores, medidas de desempenho e planejamento estratégico. A Seção 3 examina e compara algumas ferramentas comerciais e de código aberto que também propõem a implantação de BSC. A Seção 4 descreve os componentes e funcionalidades do SISCORE, bem como os resultados que ele apresenta. A Seção 5 mostra como implementar um BSC com o uso do SISCORE. Por fim são apresentadas conclusões deste trabalho.

2 FUNDAMENTOS

O objetivo desta Seção é abordar assuntos pertinentes a fundamentação utilizada para o desenvolvimento de uma ferramenta que implemente um BSC em uma instituição, tal como o SISCORE. As principais características desta metodologia dirigida ao planejamento estratégico serão apresentadas a seguir.

2.1 Balanced Scorecard

O Balanced Scorecard é uma ferramenta que busca principalmente alinhar os objetivos estratégicos através de indicadores que medem desempenho.

O Balanced Scorecard traduz a missão e a estratégia das empresas em um conjunto abrangente de medias de desempenho que serve de base para um sistema de medição e gestão estratégica [...] permite que as empresas acompanhem o desempenho financeiro, monitorando, ao mesmo tempo, o progresso na construção de capacidades e na aquisição dos ativos intangíveis necessários para o crescimento futuro (KAPLAN e NORTON, 1997, p. 2).

Kaplan e Norton (1997) buscaram mostrar como é possível medir desempenho na gestão estratégica através de indicadores não só financeiros mas, com grande relevância, os que não são monitorados e comumente registrados na contabilidade tradicional.

2.1.1 Histórico

Segundo Costa (2006) a perda da liderança em gestão e tecnologia dos Estados Unidos para o Japão motivaram as pesquisas na área da contabilidade gerencial para colocar as empresas norte-americanas novamente no topo. A contabilidade, por si só, não era mais

suficiente para acompanhar as mudanças que aconteciam nos processos de produção.

O tradicional sistema contábil de custos, baseado em premissas de produções de longo prazo de produtos padrões, sem mudanças de características e especificações, não era mais relevante no novo ambiente competitivo. A contabilidade não havia acompanhado adequadamente as mudanças. Os textos contábeis não continham um mínimo de discussão sobre técnicas, procedimentos ou propriedades da mensuração de produtividade (COSTA, 2006, p. 1).

Os criadores do BSC, Robert S. Kaplan e David P. Norton estudaram, a partir de 1990, o desenvolvimento de modelos que pudessem medir desempenho. Examinaram vários estudos de caso. Um deles, chamado *scorecard*, da empresa Analog Devices, usava medidas financeiras, medidas baseadas em prazos de entrega a clientes, medidas de ciclo de processos e medidas de eficácia de novos produtos. A este estudo de caso foram acrescentadas novas ideias que aumentaram as dimensões do *scorecard*, originando e organizando um novo método que tinha seu eixo em torno de quatro perspectivas: a financeira, a do cliente, a interna e a de inovação e aprendizado. A conclusão deste primeiro estudo ocorreu em dezembro de 1990.

Em fevereiro de 1992, Kaplan e Norton (1997) publicaram um artigo que abordava esse novo método. Muitas empresas solicitaram sua implementação. O passo seguinte foi convencer as empresas a usarem o BSC não somente como um sistema de indicadores mas aliar essas medidas à estratégia organizacional. Ao passo que cada implementação ocorria, mais o BSC se transformava em um sistema de gestão estratégica.

Atualmente muitas empresas têm implementado o BSC. O uso contínuo desse método, vem direcionando as ampliações e adaptações às novas formas de gestão. Consequentemente, ajustes tornaram-se necessários. Adaptações como adequação ao tipo de negócio, são usados também como estabelecimento de medidas padronizadas aos modelos de negócio (KAPLAN e NORTON, 1997).

Kaplan e Norton (1997) fundaram a BSCol – *Balanced Scorecard Collaborative* que tem o objetivo de facilitar o conhecimento, a utilização, o desenvolvimento e a integridade do uso da metodologia do Balanced Scorecard. A BSCol ainda oferece às organizações e indivíduos um centro de excelência, especialização e experiência em assuntos relacionados ao *Balanced Scorecard*. O BSCol presta treinamento, consultoria e certificação em BSC (PALLADIUM, 2009).

2.1.2 Conceitos

Uma ferramenta de gestão estratégica como o BSC trabalha com muitos conceitos e detalhes que devem estar claros a quem lançar mão deste recurso. O domínio desses conceitos torna-se fator primordial no desenvolvimento dessa metodologia.

2.1.2.1 *Estratégia*

Estratégia é o processo em que uma empresa se posiciona competitivamente no mercado. Esse processo, também pode ser chamado de plano de ação ou planejamento estratégico.

Uma estratégia ainda pode ser definida como “[...] a coleta, a organização e a utilização de recursos de forma a ajudar a empresa a alcançar seus objetivos, tornando-a competitiva” (COSTA, 2006, p. 11).

2.1.2.2 *Mapa Estratégico*

Um dos objetos gerados pelo BSC é o mapa estratégico. Este mapa traduz o alinhamento das estratégias e estabelece as relações de causa e efeito entre elas. O planejamento estratégico inicia pelo desenho do mapa que permite monitorar o cumprimento do plano estratégico, possui objetivos, medidas, metas e ações tornando-se um dos instrumentos de análise mais importantes em um BSC. Uma representação desse mapa sugerida por Kaplan e Norton é ilustrada na Figura 1; trata-se de um padrão comumente usado.

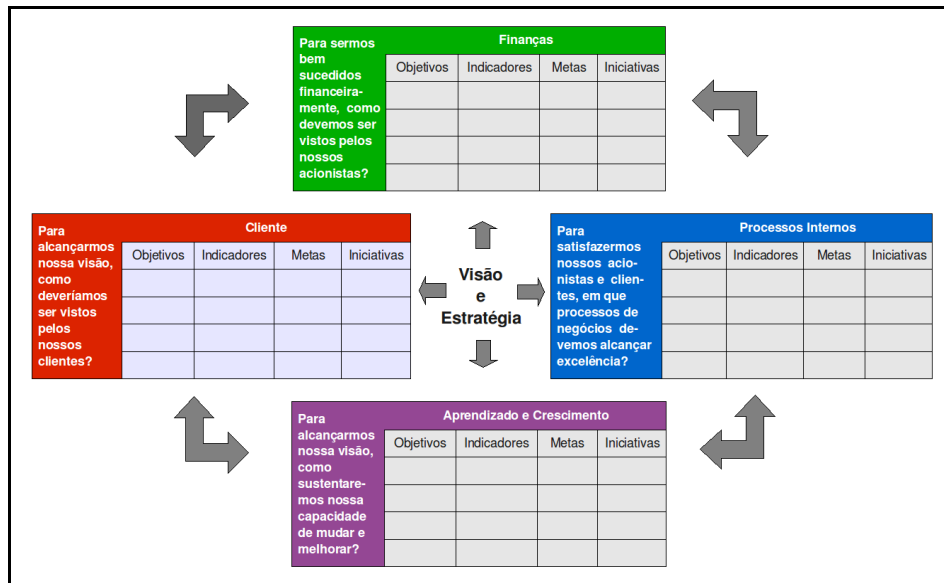


Figura 1 – Tradução da estratégia em termos operacionais

Fonte: KAPLAN e NORTON (1997, p. 10)

A estrutura mostrada na Figura 1 não é “[...] uma invenção do BSC, são sim elementos necessários de qualquer sistema de gestão de controle” (COSTA, 2006, p. 12).

As quatro perspectivas do *Balanced Scorecard* têm-se revelado adequadas em diversas empresas e setores de mercado. Mas elas devem ser consideradas um modelo, não uma camisa-de-força (KAPLAN e NORTON, 1997, p. 35).

A estrutura vista na Figura 1 pode ser ampliada com mais perspectivas como, por exemplo, ilustradas na Figura 2.

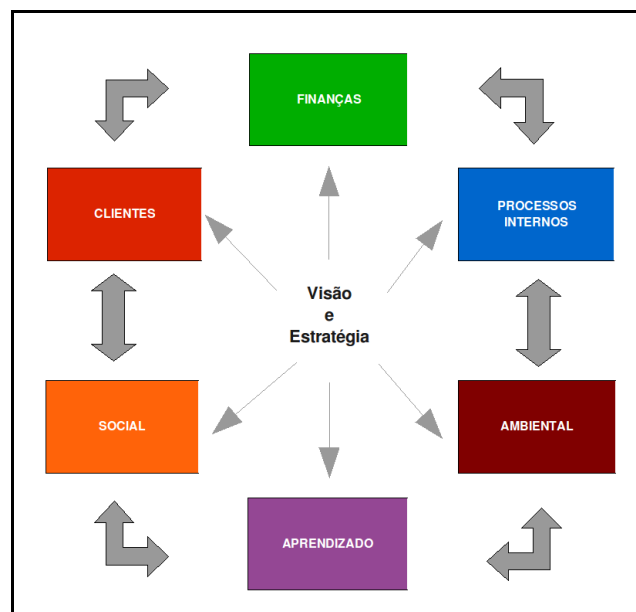


Figura 2 – Tradução da estratégia com ampliação de novas perspectiva

Fonte: O autor

As perspectivas “Social” e “Ambiental” sugeridas na Figura 2 representam aspectos atualmente relevantes na administração das empresas, mas poderiam ser quaisquer outras que também sejam importantes e que estejam alinhadas ao seus modelos de negócio. Isso fica a critério da instituição se deseja ou não adotar novas perspectivas. Algumas incorporam esses aspectos nas perspectivas tradicionais, que não é uma prática errada, entretanto não se tem um mapeamento mais preciso de áreas com aspectos tão peculiares.

2.1.2.3 Medidas de desempenho

Em um modelo contábil tradicional o processo de gerenciamento é baseado em relatórios periódicos, como balanços e balancetes, demonstrativos de resultados entre outros. Esse processo “[...] continua atrelado a um modelo contábil desenvolvido há séculos para um ambiente de transações isoladas entre entidades independentes.” (KAPLAN e NORTON, 1997, p. 7).

Inicialmente os sistemas de medidas baseavam-se essencialmente nos indicadores financeiros. As grandes empresas que surgiram durante a Revolução Industrial no século XIX trouxeram inovações nas medições do desempenho financeiro. Medidas como o Retorno sobre o Investimento (ROI) surgiram no início do século XX. Desde então o uso e a diversidade de indicadores aumentou muito (KAPLAN e NORTON, 1997).

De acordo com Moraes (2005) o processo de apresentação de resultados do BSC se dá através da medição dos indicadores de desempenho. Um indicador pode surgir de uma associação de metas numéricas pré-estabelecidas, de uma identificação do estado de um processo ou do seu resultado e, mais comumente, de uma fórmula matemática para quantificar uma medida.

De um modo geral, Moraes (2005) afirma que todas organizações possuem seus indicadores de desempenho mas que, para serem mensuráveis, devem ser facilmente levantados, compreensíveis, comparáveis e sem ambiguidades. Os indicadores são classificados como de curto, médio ou longo prazo, simples (de medição única) ou compostos, de fonte interna (Ex.: desempenho financeiro dos produtos) ou externa (Ex:

comparação com os processos dos concorrentes). Um exemplo de indicador é ilustrado no Quadro 1.

Produto/Serviço	Indicador de Desempenho	Unidade de Medida	Metas		
			2002	2003	2004
Atendimento	Satisfação do cliente	Total de reclamações			
		Total de Clientes no período	95%	98%	100%

Quadro 1 – Exemplo de indicador de desempenho

Fonte: Adaptado de Moraes (2005)

O exemplo do Quadro 1 ilustra um indicador não-financeiro. Esses indicadores ou medidas, considerados intangíveis, se devidamente explorados, podem ser mais importantes do que os tradicionais indicadores financeiros (MORAIS, 2005).

Os indicadores financeiros, segundo Costa (2006), mesmo que com inovações como o Valor Econômico Agregado (EVA), estão fundamentados sobre eventos passados. Esses tipos de indicadores são denominados *lagging* (lento, moroso). Já as medidas não-financeiras são comumente chamadas de indicadores de tendências, denominadas *leading* (condutor).

O BSC reúne e traduz ambas medidas em relatórios que direcionam a empresa a uma visão orientada a seus clientes, capacitando-a a dar respostas rápidas às mudanças de mercado, qualificando sua equipe de trabalho, agilizando o lançamento de novos produtos e tornando sua gerência capaz de projetar-se a longo prazo (COSTA, 2006).

2.1.2.4 Metas

De acordo com Costa (2006) o cumprimento dos objetivos estratégicos inicia na definição e quantificação das metas para cada medida / indicador. São as metas que definem o desempenho desejado. Este deve ser expressado a médio e longo prazos. A Figura 3 apresenta um objetivo com suas respectivas metas e valores alcançados, juntamente com uma representação gráfica dos resultados.

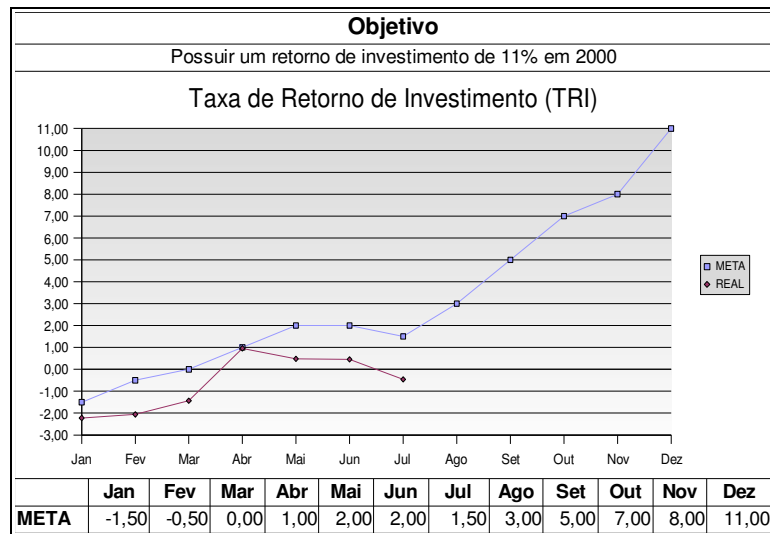


Figura 3 – Exemplo de um objetivo e suas metas

Fonte: Adaptado de Müller (2001).

Os valores do objetivo representado na Figura 3 foram obtidos através da divisão do valor lucro líquido pelo valor do ativo total respectivos a cada mês.

2.1.2.5 Plano de Ação

Um plano de ação inicia quando a construção do mapa estratégico é concluída. O esforço a ser empenhado nas medidas para alcançar as metas estabelecidas está diretamente relacionado às metas. Os planos são diferenciados de acordo com a definição dos prazos que são atribuídos às metas, ou seja, é necessário ter um plano de ação para cada meta e a estas devem ser atribuídos prazos (COSTA, 2006).

O *Balanced Scorecard* entra como uma estrutura da ação estratégica; ele inicia um importante processo na viabilização do planejamento estratégico.

2.1.2.6 O *Balanced Scorecard* na estruturação do planejamento estratégico

Para que um planejamento estratégico seja bem estruturado, além de outras, está

principalmente a medida de conduzir todos os envolvidos no processo, desde o menor até o mais alto escalão, a um esforço comum para a construção desta etapa tão importante em um BSC.

Os funcionários da linha de frente precisam compreender as consequências de suas decisões e ações; os altos executivos precisam reconhecer os vetores do sucesso a longo prazo. (KAPLAN e NORTON, 1997, p. 9).

Kaplan e Norton (1997) constataram que muitas empresas usavam medidas tanto financeiras como não financeiras. Entretanto essas medidas, sem o BSC, normalmente eram usadas para melhorias isoladas como operações com clientes, ou ainda realizadas em uma esfera administrativa superior para operacionalizar as esferas menores. Isso, no final, dava resultado somente à ações isoladas e ainda a curto prazo. O BSC explicita a necessidade de envolver todos os níveis da organização. Um modelo que representa bem essa necessidade é ilustrado na Figura 4.

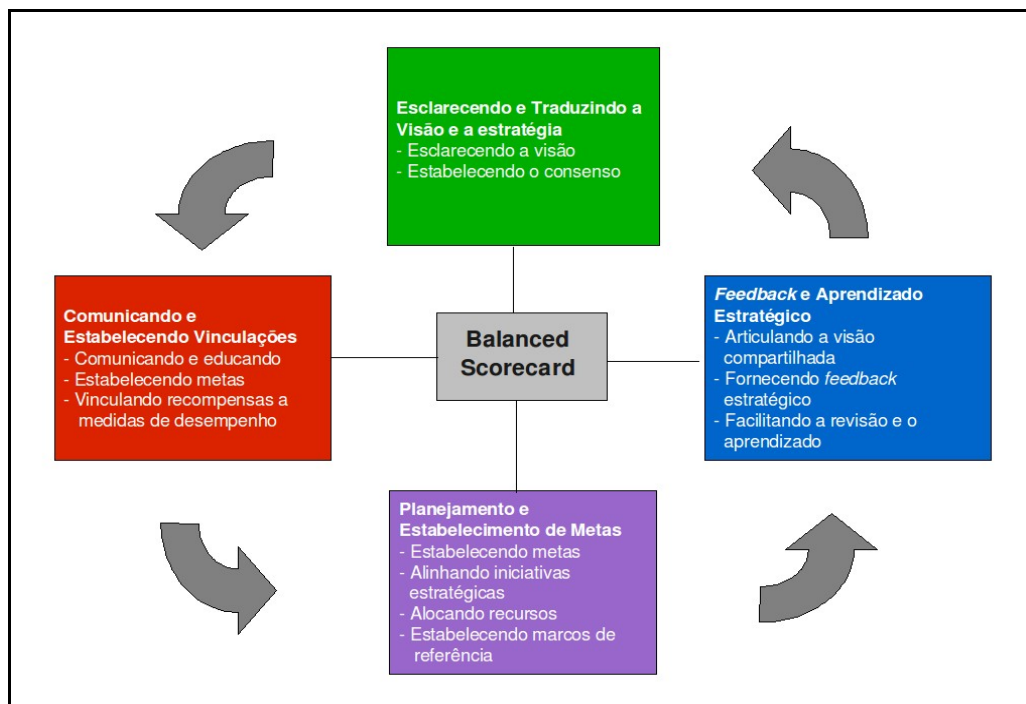


Figura 4 – Estrutura para ação estratégica no BSC

Fonte: Adaptado de Kaplan e Norton (1997).

O BSC é implementado nas empresas como um sistema de gestão estratégica e segue, habitualmente, o modelo sugerido na Figura 4. Neste exemplo, que é usado como um padrão para esta estrutura, pode-se observar as funções de cada passo no plano de ação para cada uma

das quatro perspectivas sugeridas por Kaplan e Norton (1997).

2.2 Metodologias para desenvolvimento

É importante que todo processo de análise, projeto e desenvolvimento de *software* use uma ou mais metodologias. A adoção de metodologias de desenvolvimento produz resultados melhores, uma vez que elas, na maioria dos casos, são usadas, testadas e seus resultados conhecidos. Nas próximas seções estão descritas todas as metodologias aplicadas ao desenvolvimento do SISCORE.

2.2.1 Padrão de projeto MVC

O uso de padrões de projeto no desenvolvimento de *software* estimula o uso de vocabulários comuns, auxilia na documentação e aprendizado, torna a ferramenta desenvolvida em um alvo de fácil reusabilidade e expansão e constrói um ciclo de vida saudável (GAMMA, *et al.*, 2000).

Um dos padrões de projeto mais utilizados atualmente, principalmente em aplicações Web, é o MVC – *Model, View, Controller* (Modelo, Visão e Controle). De acordo com Gamma, *et al.* (2000) a ideia é separar a aplicação (*software*) em três papéis. O Modelo é responsável pela aplicação, a Visão, responsável pela apresentação visual e o Controle que determina como a Visão reage às intervenções do usuário. A Figura 5 representa como interagem os três diferentes papéis em uma aplicação que usa o padrão de projeto MVC.

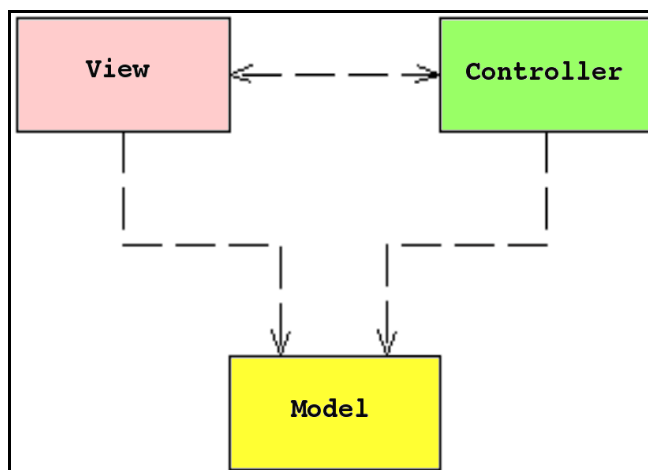


Figura 5 – Padrão de projeto MVC

Fonte: Adaptado de Fowler (2006)

O MVC representado na Figura 5 mostra como a interação da interface com o usuário é dividida em papéis (FOWLER, 2006). O uso deste padrão em aplicações Web, facilita a manutenibilidade, porque os papéis em aplicações desse tipo são distintos e as tarefas também. Por exemplo: a Visão fica a cargo, normalmente, de um *webdesigner*, que é responsável pelo aspecto e apresentação que a aplicação terá ao usuário. O Controle e o Modelo são de responsabilidade do programador. Além de todas essas vantagens, o padrão MVC tem sido amplamente utilizado em aplicações Web e proporciona bons resultados.

2.2.2 Ajax

Ajax é um acrônimo que significa *Asynchronous JavaScript And XML* (Javascript e XML assíncrono). É um conceito que reúne técnicas para navegação e atualização de páginas Web sem a necessidade de provocar sucessivos *refreshes* (SOARES, 2006).

Uma aplicação Web é composta por diversos formulários que possuem dados que devem ser validados e persistidos em um banco de dados. Essas validações e gravações ocorrem após a submissão desses formulários ao computador servidor onde é executada a aplicação. Nesse processo ocorre o que se chama de *refresh*, ou seja, sempre que a página é submetida e ocorre um retorno do computador servidor, existe o carregamento de uma nova página e o *browser* (navegador) faz um reprocessamento da página que existia antes da sua

submissão. Isso não é agradável visualmente. A proposta do uso de técnicas de Ajax é diminuir esse desconforto e melhorar a aparência e funcionamento das aplicações Web, dando um aspecto de aplicação *desktop*.

2.2.3 Orientação a objetos

Desde que se produz *software* têm-se aprimorado as técnicas de programação com o intuito de melhorar performance, qualidade, segurança, legibilidade do código, padronização, entre outros. Esses processos, no passar do tempo, sempre sofreram mudanças. Houve um marco importante na forma de produzir *software* quando foi introduzido o paradigma de Orientação a Objetos (OO).

Orientação a Objetos, ou Programação Orientada a Objetos (POO), é uma forma diferente de conceber uma aplicação. Esse paradigma aborda fortes conceitos de reusabilidade de código e mudou totalmente a forma de projetar e escrever programas.

Antes os programas possuíam funções que desempenhavam determinadas tarefas; em POO existem objetos que possuem características, comportamentos e trocam mensagens com outros objetos. Isso revolucionou a indústria do *software* e possibilitou a criação de padrões, métodos e ferramentas para auxiliar e facilitar a tarefa de programar.

Atualmente, existe uma quantidade grande de linguagens de programação que usam esse novo paradigma, inclusive a linguagem utilizada para o desenvolvimento do SISCORE, que é a PHP 5.

2.2.4 Processo de *software*

De acordo com Pressman “[...] a elaboração de *software* de computador é um processo interativo de aprendizado, e o resultado [...] um conhecimento personificado acumulado, destilado e organizado a medida que o processo é conduzido” (PRESSMAN, 2002, p. 17).

Um processo de *software* é um roteiro necessário para construção de *software* de alta qualidade. Os processos de *softwares* são divididos em modelos que incorporam estratégias de desenvolvimento divididas em fases. (PRESSMAN, 2002). Nas próximas seções são apresentados os principais modelos de processo para engenharia de *software*.

2.2.4.1 Modelo sequencial linear

Também chamado de ciclo de vida clássico ou modelo em cascata. Segundo o próprio nome sugere, aborda o desenvolvimento de *software* de forma sistemática sequencial. É um modelo utilizado quando os objetivos, funcionalidades e resultados desejados são conhecidos.

Essa abordagem inicia estabelecendo-se os requisitos para todos os elementos do sistema, definindo todas as interações que ele possa contemplar, seja entre outros *softwares* e e suas interações com elementos como itens de *hardware*.

O ciclo continua com a definição e análise dos requisitos focalizado-os no *software*. Esses requisitos devem ser documentados e revistos com o cliente. Após isso, inicia o projeto de *software* que é composto pela definição da estrutura dos dados, arquitetura do *software*, seu comportamento, desempenho e interface.

Concluída essa etapa, efetua-se a codificação, que é a escrita e geração de código para a linguagem de máquina. A medida que os códigos vão sendo gerados já é possível iniciar a fase de testes que busca descobrir erros e garantir que todas as funcionalidades produzam os resultados desejados.

Por fim, a fase de manutenção que modifica o *software* acrescentando ou modificando

funcionalidades e correção de erros não previstos ou não encontrados na fase de testes. (PRESSMANN, 2002).

2.2.4.2 Modelo de prototipagem

O modelo de prototipagem se adapta bem em situações onde o cliente não identifica os requisitos de entrada, processamento ou saída, mas define um conjunto de objetivos e em outras situações semelhantes.

O ciclo de um modelo de prototipagem inicia pela definição dos requisitos encontrando e definindo os objetivos gerais. Isso identifica as necessidades conhecidas e apontam as áreas que precisam se demarcadas.

A partir disso um projeto rápido é realizado quando se concentra na representação os aspectos que ficarão visíveis ao cliente. Isso se dá através de protótipos muitas vezes não funcionais no início e progredindo para protótipos cada vez mais funcionais a medida que as necessidades e requisitos são definidos.

Esta é uma abordagem que agrada tanto a clientes como a desenvolvedores, pela rapidez que os resultados, mesmo que incompletos, são apresentados.

Resumindo, esse paradigma consiste em (i) Ouvir o cliente, (ii) Construir / revisar o protótipo, (iii) Entregar o protótipo para o cliente testar, retornando ao primeiro passo novamente até que a aplicação atenda às necessidades do cliente (PRESSMANN, 2002).

2.2.4.3 Modelo RAD

O desenvolvimento rápido de aplicações (*Rapid Application Development*) possui a ênfase voltada a ciclos de desenvolvimento incrementais muito curtos. Esse modelo foi adaptado do modelo sequencial linear. Um projeto é candidato a esse modelo quando a aplicação pode ser modularizada e desenvolvida em menos de 3 meses.

A construção do *software* é baseada em componentes e a rapidez no desenvolvimento também está em contar com uma equipe que compreenda plenamente os requisitos e restrinja os objetivos. (PRESSMANN, 2002).

2.2.4.4 Modelo de *software* evolucionário

Projetos que sofrem frequentes mudanças e evoluções a medida que são desenvolvidos podem enquadrar-se no modelo de *software* evolucionário, por serem interativos e permitirem o desenvolvimento de aplicações cada vez mais completas.

Os modelos incremental e espiral são do tipos evolucionário. O espiral combina elementos do modelo sequencial linear e da prototipagem, já o modelo espiral se desenvolve junto a uma série de versões incrementais. (PRESSMANN, 2002).

2.3 Tecnologias para Desenvolvimento de Aplicações Web

O desenvolvimento de aplicações Web necessita do uso conjunto de diversos recursos como linguagens de programação, banco de dados, esquemas para tráfego de dados a estilos para visualização das informações.

Existem diversas tecnologias que são utilizadas na implementação de aplicações direcionadas para Web. A gama de soluções livres e de código aberto é vasta, tanto no que se refere a opções para implementação quanto na qualidade das suas documentações. Entretanto, faz-se necessário optar pela que melhor se adapta às necessidades das soluções propostas, independente de plataforma, tanto de *hardware* como de *software*, e que disponha de documentação rica e uma comunidade suficientemente ativa para dar suporte à demanda de aplicações de porte corporativo como proposto por este trabalho. As seções a seguir descrevem todos os recursos que foram usados no desenvolvimento do SISCORE.

2.3.1 Linguagens

Dentre as diversas linguagens disponíveis para desenvolvimento de aplicações, existem as voltadas especificamente para Web. A escolha por uma opção surge quando são definidos os objetivos e resultados esperados. Essa seleção não se restringe a uma linguagem específica, mas na combinação mais coerente, baseada na interação existente entre as diferentes tecnologias que serão aplicadas.

2.3.1.1 PHP

A linguagem de programação PHP (PHP, 2009), denominada como *Hipertext Preprocessor*, teve seu lançamento em 1995 por Rasmus Lerdorf. Sua primeira versão surgiu como uma coleção de *scripts*⁵ chamada de PHP/FI (*Personal Homepage Tools / Form Interpreter*). Assemelhava-se à linguagem Perl⁶ e era usada principalmente para submissão de formulários via Internet e a manipulação dos dados dos mesmos. (GUTMANS, BAKKEN e RETHANS, 2005).

Gutmans e Suraski (2005) começaram fazer alterações para comportar mais funcionalidades criando, em 1997, o PHP/FI 2. Logo iniciaram a reescrita de todo código e, junto com seu criador Rasmus Lerdorf, lançaram, em 1998, a versão 3 que suportava, entre outras funcionalidades, acesso a banco de dados. Este foi o início da popularização da linguagem de programação PHP.

Atualmente PHP está na versão 5 e espalhou-se por todo o mundo. Opera em mais de um terço dos servidores Web. Conta com muitas funcionalidades das quais se destaca o suporte a orientação a objetos (GUTMANS, BAKKEN e RETHANS, 2005).

⁵ Arquivos contendo comandos da linguagem executada na aplicação.

⁶ *Practical Extracting and Reporting Language* – linguagem de programação prática para extração de informações de arquivo de texto e geração de informações a partir de conteúdo de arquivos

2.3.1.2 HTML

De acordo com Raggett (2007) HTML - *HyperText Markup Language* (Linguagem de Marcação de Hipertexto) é um tipo especial de texto utilizado por *browsers* para apresentação de textos e gráficos. Este texto integra um conjunto de *tags* (marcas) que definem a forma de apresentação dos textos, figuras e outros elementos nas páginas Web.

Desde a sua criação, na década de 1990, foram incorporadas diversas modificações que padronizaram esse tipo de programação. Atualmente está na versão 4.0.1 e é regulamentado pelo W3C - *World Wide Web Consortium*⁷.

2.3.1.3 CSS

Budd, Moll e Collinson (2006) relatam que a Web, inicialmente, era composta por uma série de documentos formatados em HTML, entretanto, com o aumento do uso da Internet, houve a necessidade de acrescentar funcionalidades aos *sites*. As funcionalidades que o HTML proporcionava não eram suficientes para demanda por recursos mais avançados. Ao passo que o aspecto das páginas tornava-se cada vez mais importante, a leitura das marcações HTML nos códigos das páginas tornou-se cada vez mais difícil, pois tinha uma estrutura pouco compreensível.

O uso de CSS - *Cascading Style Sheets* (folhas de estilo em cascata), surgiu para separar a aparência externa da página do seu conteúdo, ou seja, elementos visuais como cor, tamanho e tipo da fonte são tratados no código CSS enquanto que seu conteúdo, no código HTML. Isso foi um passo importante na melhora na inteligibilidade dos códigos que, por sua vez, trouxe muitos benefícios para o desenvolvimento de páginas Web, tornando a marcação significativa e de fácil entendimento (BUDD, MOLL e COLLISON, 2006).

⁷ Órgão que desenvolve normas e padrões para a criação e interpretação de conteúdos para a Web (<http://www.w3c.org>).

2.3.1.4 XML

XML é o acrônimo de *Extensive Markup Language* – Linguagem de Marcação Extensível. É um formato de texto simples e flexível concebido para a troca de informações entre a grande variedades de dados existentes na Web (W3C, 2007).

O objetivo é representar qualquer informação através de uma linguagem simples e única, de modo que todos os aplicativos possam trocar informações entre si, formando, assim, uma camada de comunicação universal.

2.3.1.5 Javascript

Javascript é uma linguagem que foi criada inicialmente com o nome de LiveScript. Foi desenvolvida na Netscape⁸, para tornar seu *browser* mais funcional. Uma das principais funcionalidades é a validação de campos de um formulário, cálculos, criação de novas janelas exibindo mensagens ou solicitando algumas informações ao usuário, tarefa que antes eram executadas no servidor, causando inúmeros *refreshes*. Com Javascript parte do processamento é executado no lado do cliente, ou seja, no computador onde estão sendo exibidas e solicitadas as informações (SOARES, 2006).

2.3.1.6 Banco de dados

Um Sistema de Gerenciamento de Banco de Dados (SGBD) é projetado para armazenar e gerenciar informações. Em um banco de dados as informações são guardadas com estruturas e formas de manipulação específica a cada propósito de uso. Essas estruturas e

⁸ Uma das primeiras empresas a disponibilizar um navegador para internet (<http://www.netscape.org>).

mecanismos de controle e armazenamento de informações possuem dispositivos especiais de segurança que prezam sua integridade e confidencialidade. Os Sistemas Gerenciadores de Banco de Dados, através de um conjunto de programas, controlam todos esses dados e funcionalidades (SILBERSCHARTZ, KORTH e SUDARSHAN, 1999).

As informações são armazenadas em um banco de dados em forma de tabelas cujas colunas representam os campos de um registro e cada linha representa um registro. A forma de acessar, modificar e inserir as informações no banco de dados é por meio da linguagem SQL - *Structured Query Language* (Linguagem de Consulta Estruturada).

No mercado existem diversos SGBDs tais como MySQL⁹, PostgreSQL¹⁰, Oracle¹¹, entre outros. Importa ressaltar que uma aplicação inteligente deve deixar a cargo do cliente escolher qual SGBD usar, ou seja, o *software* deve operar independente de plataforma de banco de dados, ou, pelo menos, estar preparado para funcionar com os principais SGBD existentes. Isso também é contemplado pelo SISCORE.

A vantagem do cliente poder optar por qual banco de dados usar é a flexibilidade que ele tem para dimensioná-lo ao negócio, pois existem SGBDs mais robustos, como PostgreSQL e Oracle, que são projetados para gerenciar um volume de informações muito grande, enquanto existem outros de porte menor, como SQLite¹² ou Firebird¹³. Dimensionar um banco de dados é muito importante, pois ganha-se desempenho se esta tarefa for bem feita.

2.3.1.7 UML

UML – *Unified Modelling Language* (Linguagem de Modelagem Unificada) é uma forma de representação de um sistema cujo paradigma usado é o de modelagem com orientação a objetos.

⁹ MySQL (<http://www.mysql.org>)

¹⁰ PostgreSQL (<http://www.postgresql.org>)

¹¹ Oracle (<http://www.oracle.com>)

¹² SQLite (<http://www.sqlite.org>)

¹³ Firebird (<http://www.firebirdsql.org>)

A UML foi projetada para auxiliar aqueles que participam da atividade de desenvolvimento de *software* a construir modelos que permitam visualizar o sistema, especificar a estrutura e o comportamento deste, construí-lo e documentar as decisões tomadas durante o processo (SCOTT, 2003, p. 19).

Scott (2003) explica que UML tem origem no *Rational Unified Process* – RUP (Processo Unificado da Rational). A Rational é uma empresa que comprou a ideia desenvolvida no trabalho de Ivar Jacobson, na empresa Ericson, no final da década de 1960. Essa ideia surgiu da necessidade de modelar um sistema muito grande de telecomunicações.

Posteriormente foi incorporada à engenharia de *software*, que em 1987, deu origem à companhia Objectory AB. Em 1995 foi publicado o livro *Object-Oriented Software Engineering* (Engenharia de *Software* Orientado a Objeto). Pouco depois a Rational comprou a Objectory AB que expandiu o método e originou a UML (SCOTT, 2003). Mas, segundo Fowler e Scott (2000), o Processo Unificado é distinto de UML, ou seja, não é necessário usar RUP para usar UML.

2.3.2 Frameworks

Frameworks formam uma base para o desenvolvimento de um projeto maior. “[...] é uma coleção de códigos fonte, classes, funções, técnicas e metodologias que facilitam o desenvolvimento de novos *softwares*.” (MINETTO, 2007, p. 17).

O desenvolvimento de uma aplicação com a ajuda de um *framework* tende a resolver problemas que normalmente demandam bastante tempo se desenvolvidos da maneira tradicional. As soluções encontradas vão desde gerar automaticamente uma aplicação com o padrão MVC a tratar automaticamente o acesso a banco de dados sem que seja necessário escrever uma linha de SQL. O SISCORE é um *software* desenvolvido usando o padrão de projeto MVC. Para aplicar esse padrão existem diversas opções de *frameworks* que proporcionam uma base para o desenvolvimento de um projeto maior.

2.3.2.1 Comparação entre Frameworks

A escolha de um *framework* foi fundamental para o projeto de implementação do SISCORE, uma vez que ele deveria atender às necessidades de implementação e proporcionar resultados de desempenho satisfatório usando tecnologias e metodologias atuais.

A tarefa de escolher um *framework* se deu através da identificação das características de cada um e, como fator decisório, o estudo apresentado através de um *benchmark*¹⁴ que comparou os três mais bem colocados (CAKEPHP Brasil, 2009). O Quadro 2 apresenta a relação dos principais e mais conhecidos *frameworks* usados atualmente, bem como suas principais características.

Framework	Funcionalidades (vide Quadro 3)											
	PHP4	PHP5	Múltiplos BD's	ORM	DB Objects	Templates	Caching	Validation	Ajax	Auth Module	Modules	EDP
Yii		Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
CodeIgniter	Sim	Sim	Sim		Sim	Sim	Sim	Sim				
CakePHP	Sim	Sim	Sim	Sim	Sim		Sim	Sim	Sim	Sim	Sim	
PHPDevShell		Sim				Sim			Sim	Sim	Sim	
Akelos	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	
Prado		Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Symfony		Sim	Sim	Sim	Sim		Sim	Sim	Sim	Sim	Sim	
Zend		Sim	Sim	Sim	Sim		Sim	Sim	Sim	Sim	Sim	
Zoop	Sim	Sim	Sim		Sim	Sim	Sim	Sim	Sim	Sim		
Fuzebox	Sim	Sim	Sim				Sim		Sim		Sim	
PHP on TRAX		Sim	Sim	Sim	Sim			Sim	Sim		Sim	
QPHP	Sim	Sim	Sim		Sim	Sim		Sim	Sim	Sim	Sim	Sim
Seagull	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	

Quadro 2 – Relação de *frameworks* e as funcionalidade que eles atendem

Fonte: Adaptado de Phpframeworks (2009)

A descrição das funcionalidades apresentadas no Quadro 2 são apresentadas no Quadro 3. Observa-se no Quadro 2 que os *frameworks* apresentam diferentes funcionalidades e que alguns como o Yii e o Zend dão suporte somente a PHP5, outros como o CakePHP e o CodeIgniter dão suporte a PHP4 também. Entretanto existem diferenças importantes que dão peso na decisão optar por um deles. Como no caso do CakePHP que dá suporte ao PHP4 e PHP5 mas é desenvolvido em PHP4.

¹⁴ Aplicação de um padrão de testes para avaliar performance.

Funcionalidades	Significado
Múltiplos BD's	Suporta múltiplos bancos de dados[]
ORM	Suporta ORM ¹⁵ usualmente chamado de <i>ActiveRecord</i> ¹⁶
DB Objects	O <i>framework</i> permite inclusão de objetos de outros bancos de dados
Templates	Implementa a camada de <i>view</i> do MVC através de <i>templates</i> ¹⁷
Caching	Possui <i>cache</i> ou formas de reproduzir <i>cache</i> ¹⁸
Validation	Possui componentes de validação e filtragem de dados
Ajax	Possui suporte a Ajax
Auth Module	Possui módulo de autenticação de usuários
Modules	Possui suporte a utilidades tais como <i>feeds</i> RSS ¹⁹ , geração de PDF, etc, comumente chamados de <i>helpers</i> (auxiliadores)
EDP	Indica que possui suporte a <i>Event Driven Programming</i> ²⁰

Quadro 3 – Descrição das funcionalidades citadas no Quadro 2

Fonte: o autor

As funcionalidades descritas no Quadro 3 são características de recursos e padrões largamente utilizados em aplicações Web. O mesmo *site* apresenta um *ranking*²¹ com os 10 *frameworks* mais usados, esses dados estão relacionados no Quadro 4.

Colocação	Framework	Grau de Avaliação
1	Yii	4,43
2	CakePHP	4,21
3	CodeIgniter	4,2
4	PHPDevShell	4,14
5	Akelos	4,05
6	Prado	4
7	Symfony	3,84
8	Zend	3,8
9	ZooP	3,6
10	QPHP	2,33

Quadro 4 – Colocação dos *frameworks* mais baixados até 25 de março de 2009

Fonte: Adaptado de Phpframeworks (2009)

¹⁵ *Object-Relational Mapping* (Mapeamento Objeto-Relacional) é uma técnica desenvolvida para representar as tabelas do banco de dados como classes dentro de um *software* orientado a objetos.

¹⁶ Padrão que implementa o ORM.

¹⁷ Modelos / formas de apresentação visual dentro do padrão de projeto MVC.

¹⁸ Dispositivo ou área de armazenamento rápido.

¹⁹ Mecanismos para receber atualizações de sites que mudam seu conteúdo com frequência.

²⁰ Programação Orientada a Eventos é baseada no paradigma onde fluxo é determinado pelos acontecimentos como cliques do mouse, pressionamento de determinadas teclas, etc.

²¹ Posicionamento de um item dentro de uma escala de classificação baseada em resultados de uma estatística.

O Quadro 4 apresenta o Yii, o CakePHP e o CodIgniter como os três primeiros, assim como no Quadro 2 esta ordem foi mantida para que se possa comparar os recursos em função da sua colocação. Um *framework* deve ter vários predicados para torná-lo um produto de boa qualidade, mas existem alguns desses que se destacam mais, tais como segurança, integridade dos dados, boas práticas de programação através de padrões de projeto, robustez e desempenho.

Os Gráficos 1 e 2 e 3 mostram um *benchmark* entre três *frameworks*. Os resultados apresentados nesses gráficos foram obtidos através do uso da ferramenta AB (*Apache HTTP server benchmarking tool*)²² em um computador com processador Intel Xeon 2.66GHz e 256MB de RAM com sistema operacional Debian Etch e Apache 2.2.3, PHP 5.2.0 e MySQL 5.0.32 instalados.

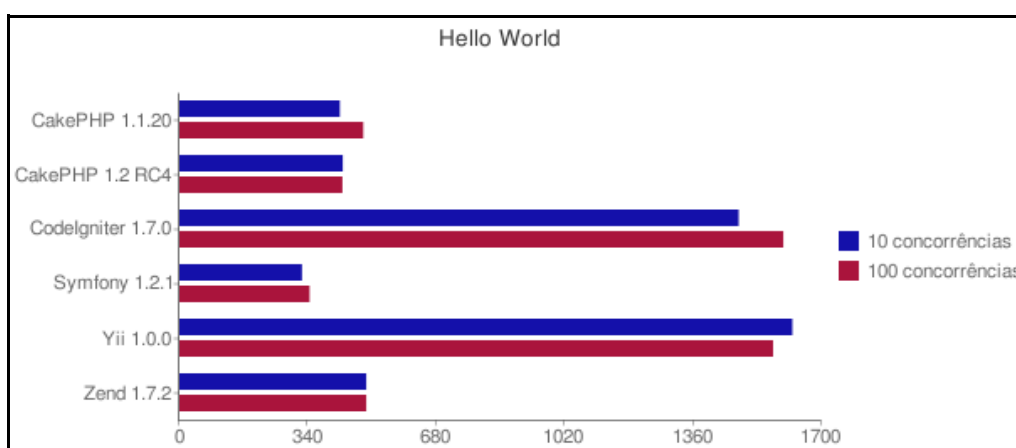


Gráfico 1 - Benchmark de frameworks com 10 e 100 concorrências com programa “Hello World”

Fonte: Cakephp Brasil (2009)

No Gráfico 1 estão os resultados de testes com um programa “*Hello World*” com 10 e 100 requisições a cada 30 segundos. No eixo Y estão representadas o número de requisições completadas a cada teste. Observa-se que os melhores desempenhos encontram-se no CodeIgniter e no Yii, com uma pequena inversão de desempenho no número de ocorrências.

A primeira vista um teste com esse parece muito pouco com uma aplicação do mundo real. Entretanto grande parte dos aplicativos Web apresentam-se em situações muito semelhantes a um simples “*Hello World*”, ou seja, em casos onde páginas possuem grande parte de seu conteúdo armazenado em cache; a aplicação precisa apenas buscar esse conteúdo e exibi-lo (YII, 2009). No Gráfico 2 está representado um teste com acesso a banco de dados

²² AB – (<http://httpd.apache.org/docs/2.0/programs/ab.html>)

com 10 registros.

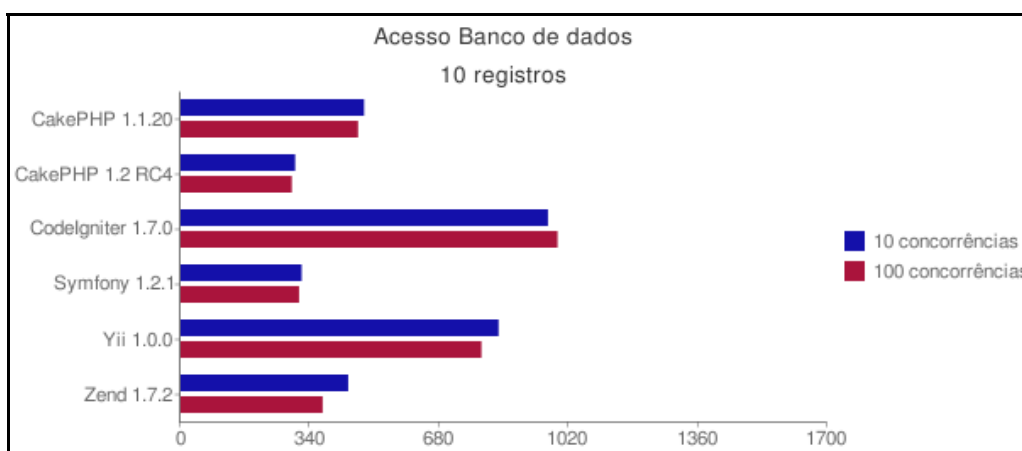


Gráfico 2 - Benchmark de *frameworks* com 10 e 100 concorrências com acesso a 10 registros em um banco de dados

Fonte: Cakephp Brasil (2009)

Pode-se observar que o desempenho mostrado no Gráfico 2 é muito semelhante ao do Gráfico 1, porém o CodeIgniter apresenta um desempenho melhor. Há um detalhe curioso entre o desempenho das duas versões apresentadas do Cake, que se mostra melhor em sua versão menos recente. Mas essa diferença desaparece quando o número de registros sobe para 1000 como apresentado no Gráfico 3.

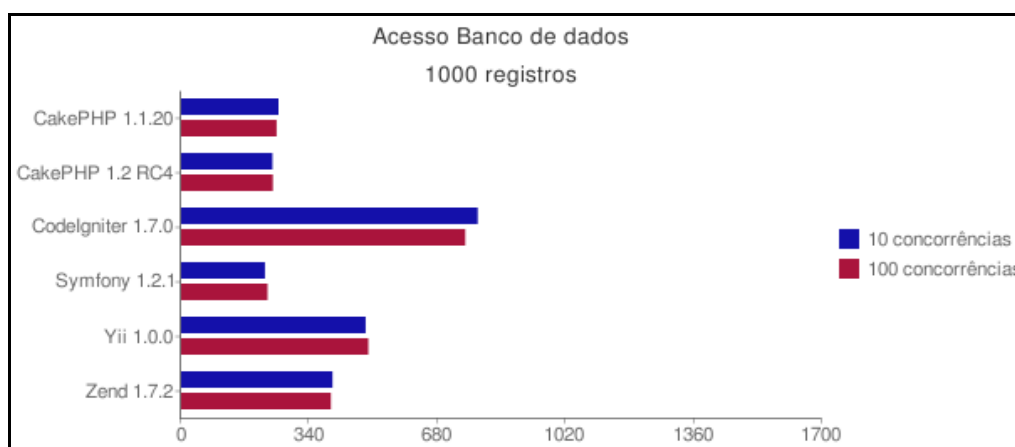


Gráfico 3 - Benchmark de Frameworks com 10 e 100 requisições com acesso a 1000 registros em um banco de dados)

Fonte: Cakephp Brasil (2009)

A diferença dos dados apresentados do Gráfico 2 para Gráfico 3 são apenas a quantidade de registro utilizados, neste caso, 1000. Nota-se que o desempenho do Yii não

manteve a proporção do Gráfico 2, mas todos apresentaram um *overhead*²³ grande. Entretanto, a não ser em relatórios, não se faz requisições a páginas com 1000 registros e quando isso ocorre deve-se aplicar a técnica de paginação²⁴ para evitar o *overhead*.

2.3.2.2 Conclusões das Comparações entre Frameworks

Cruzando os gráficos e tabelas comparativos até aqui apresentados obtém-se os dados do Quadro 5. Analisando esses dados, observa-se que, quanto ao desempenho, o Quadro 4 se inverte, colocando o CodeIgniter em primeiro lugar, entretanto deve-se pesar o fato de que ele não possui recursos de relevância importante, como suporte a Ajax, ORM e módulo de autenticação.

Item de avaliação	1	2	3
Colocação Tabela [2]	Yii	CakePHP	CodeIgniter
Funcionalidades	Yii (11)	CakePHP(10)	CodeIgniter(7)
Desempenho Gráfico (1) Hello World	CodeIgniter	Yii	CakePHP
Desempenho Gráfico (2) BD 10 Registros	CodeIgniter	Yii	CakePHP
Desempenho Gráfico (3) BD 1000 Registros	CodeIgniter	Yii	CakePHP

Quadro 5 – Comparação do desempenho entre *frameworks*

Fonte: O autor

O CakePHP é um *framework* que possui características de robustez importantes, mas tem todo seu código fonte em PHP4 o que o torna defasado, considerando que o PHP 6 já está prestes a ser lançado (GOOD, 2009). Considerando o uso de tecnologias mais recentes, um número expressivo de funcionalidades, padrões de projeto, simples e extensível²⁵, o *framework* Yii é a melhor opção, pois apresenta e aplica as melhores práticas para o desenvolvimento de aplicações Web atualmente.

²³ Excesso de processamento ou armazenamento que diminui o desempenho processamento.

²⁴ Método usado para subdividir um processo de consulta a banco de dados para evitar *overhead*.

²⁵ O *framework* Yii permite a criação de extensões que, normalmente, são disponibilizadas livremente na Internet.

3 TRABALHOS CORRELATOS

Esta Seção aborda as principais iniciativas que possuem um escopo semelhante ao do SISCORE. Foram pesquisadas algumas propostas que abordam o tema ferramenta de *software* livre para BSC e, mediante essa pesquisa, são apresentados os resultados obtidos com experiências com as ferramentas escolhidas.

3.1 Ferramentas de BSC Específicas

As ferramentas específicas para BSC, são desenvolvidas com o único objetivo de implementar os métodos desenvolvidos por Kaplan e Norton (1997). A maioria dos aplicativos que são livres foram pesquisados e obtidos no SourceForge (SOURCEFORGE.NET, 2009). Existem muitos projetos publicados e os que disponibilizam *download* estão descontinuados há bastante tempo, ou ainda não foram desenvolvidos. Não foi encontrada ferramenta de BSC que apresentasse a proposta do SISCORE e que mantenha suporte e continuidade atualmente.

O Quadro 6 mostra os principais *softwares* de BSC disponibilizados no SourceForge. A coluna “*Rank*” representa o pico de *downloads* em um determinado período e representa sua atividade; a coluna registro é a data que foi registrado no SourceFourge e *Downloads* é a coluna que indica quantas vezes cada *software* foi baixado.

Nome	Rank	Registro	Última alteração	Downloads
BSPG, an opensource Balanced Scorecard	26,72	13/05/03	24/01/04	11077
OpenScorecard	51,78	25/06/04	10/07/04	3727
BlueOxygen Relief	11,18	03/08/07	07/08/07	1486

Quadro 6 – *Softwares* de BSC disponíveis no SourceForge

Fonte: Adaptado de SourceForge (2009)

3.1.1 BSPG

BSPG (2009) é um *software* livre desenvolvido sob a licença GPL²⁶. Sua origem é italiana. Foi desenvolvido em PHP e JavaScript, o banco de dados é PostgreSQL. É uma aplicação exclusivamente Web. A maior parte das regras de negócio se concentram no banco de dados, não dando opção para uso com outro SGBD. A Figura 6 mostra como a ferramenta relaciona os objetivos entre as perspectivas.

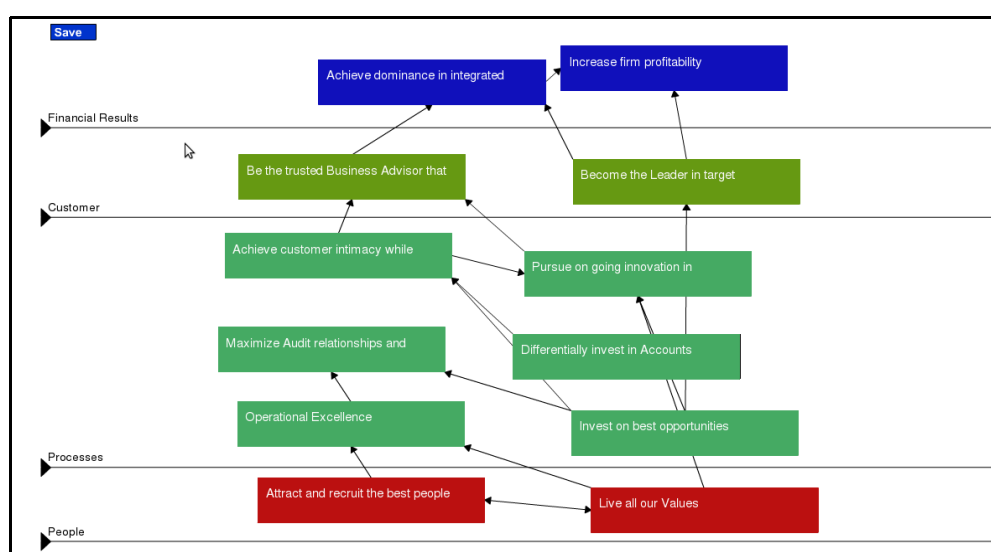


Figura 6 – Objetivos por perspectiva

Fonte: BSPG (2009)

Observa-se que o padrão da Figura 6 segue a metodologia de Kaplan e Norton (1997), representando o mapa estratégico e a relação entre os objetivos nas diferentes perspectivas, quais sejam Financeira, Clientes, Processos e Pessoas.

A Figura 7 mostra o desempenho de uma medida em três datas distintas. Cada data possui o objetivo a ser atingido o resultado até aquela data. Os dados são apresentados tanto em gráfico de barras como em forma de tabela numérica. Nota-se que quando é pressionado o botão “*Export to Excel*”, que teria por objetivo exportar os dados para uma planilha de cálculo eletrônica, a informação que se obtém é que esta uma página de demonstração estática, ou seja, não há processamento nos dados apresentados.

²⁶ *General Public Licence* – é a licença usada pela maioria dos projetos de *software* livre.

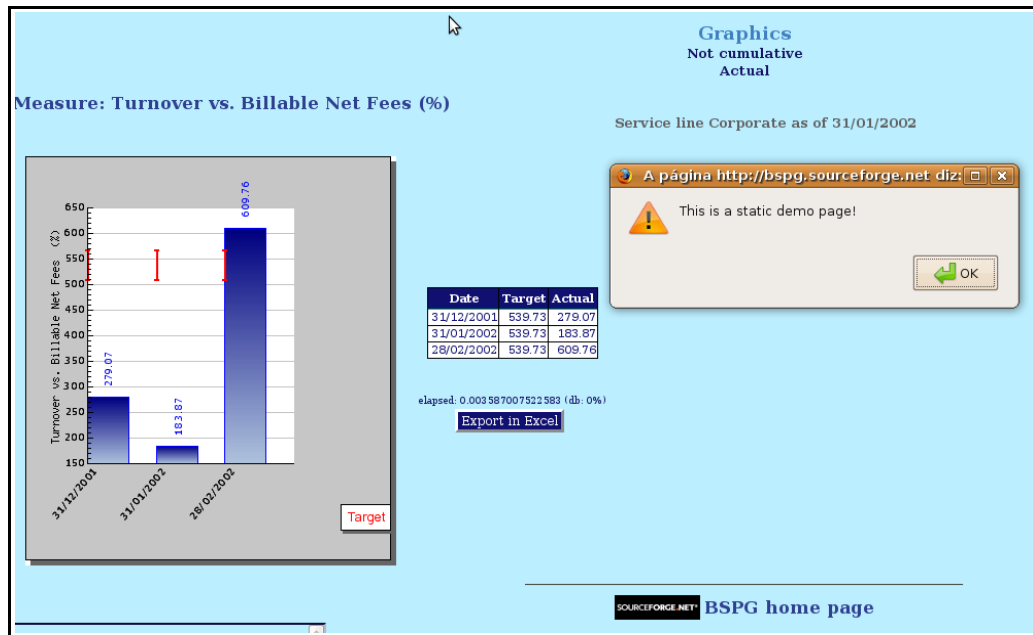


Figura 7 - Desempenho de uma medida

Fonte: BSCG (2009)

O BSPG registrou-se no SourceForge em 13 de maio de 2003 como versão 4-Beta. O projeto, apesar de ter uma quantidade expressiva de *downloads*, foi abandonado no ano de 2003. O sistema inteiro é baseado em funções e não é orientado a objeto, não existe uma classe sequer. Sua interface é pouco amigável. As funções da carga inicial do banco de dados gera várias tabelas e funções mas apresentam diversos erros. Portanto, a avaliação foi efetuada nas telas de demonstração estáticas disponíveis no site do desenvolvedor. Os dados gerados estavam inconsistentes e a versão de demonstração oferecida não estava disponível no momento desta análise.

3.1.2 OpenScorecard

OpenScorecard é uma ferramenta desenvolvida em Python²⁷. Em seu pacote disponibilizado no SourceForge o arquivo de instalação contém instruções cujos caminhos levam a *links* quebrados. Sua documentação é insuficiente, ineficiente e não possui nenhuma demonstração. Portanto, não foi possível instalar e testar suas funcionalidades. O projeto não

²⁷ Linguagem de programação orientada a objeto (<http://www.python.org>)

apresenta movimentação desde julho de 2004.

3.1.3 BlueOxygen

É um *framework* para Web desenvolvido na linguagem de programação Java, que possui em seu escopo o desenvolvimento de um BSC que se chama Relief. Em seu site os itens relativos ao BSC não há conteúdo algum. Não há documentação e não existe nenhuma demonstração. Também não foi possível instalar e testar suas funcionalidades (BLUEOXYGEN, 2009).

3.1.4 Strategic Adviser

O Strategic Adviser – SA (INTERACT, 2007) é uma ferramenta comercial, de código fechado, ou seja, proprietário. Foi desenvolvida pela Interact Solutions para controle da gestão estratégica, operacional e da qualidade. Não é livre e é comercializada em três versões distintas, quais sejam, *Enterprise Edition*, *Standard Edition* e *Academic Edition*. A primeira é aplicada a empresas de grande porte e é disponibilizada de forma completa. A segunda é uma versão reduzida, apresentando-se como uma versão direcionada para empresas que estão iniciando uma gestão administrativa. Por fim, a terceira, baseada na versão *Standard Edition*, busca integrar-se em ambientes acadêmico para fins didáticos e de laboratório. O uso desta versão é monitorada. A Figura 8 pode-se ver a o planejamento estratégico da empresa fictícia Alfa Corporation e suas quatro perspectivas segundo modelo inicialmente proposto por Kaplan e Norton (1997) na publicação do BSC em 1992.

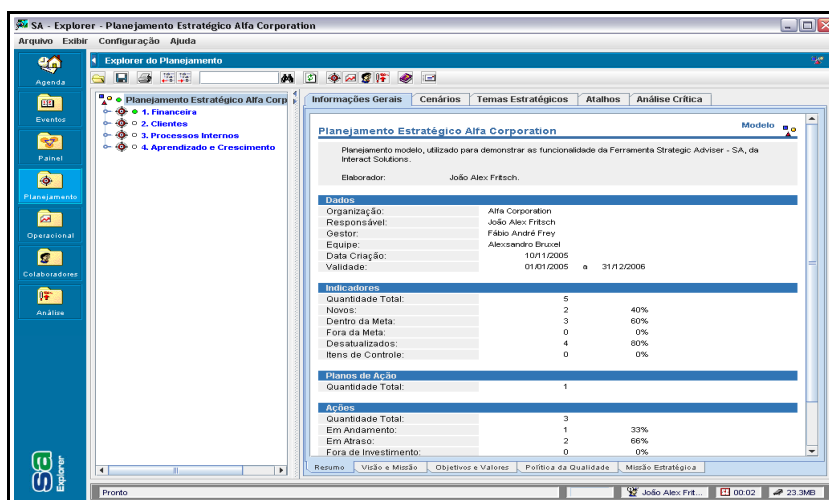


Figura 8 – Planejamento estratégico

Fonte: Interact (2007, p. 7)

Na tela, apresentada na Figura 8, são informados todos os dados do planejamento, desde informações gerais, cenários, temas estratégicos, atalhos e análise crítica, bem como sua visão, missão, objetivos, política da qualidade e missão estratégica.

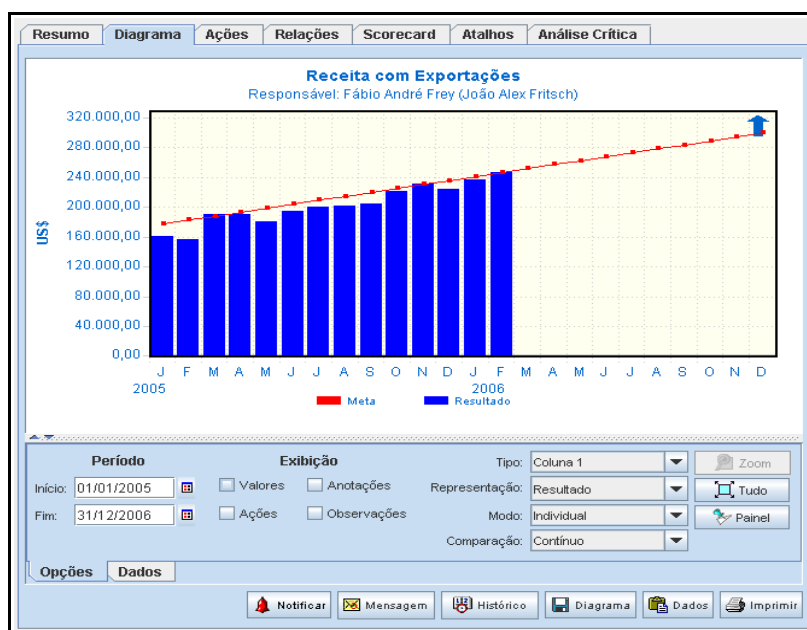


Figura 9 – Indicador e seu desempenho

Fonte: Interact (2007, p. 12)

O gráfico apresentado na Figura 9 mostra o indicador “receita com exportações” que foi gerado pela aplicação através do SA. Esse gráfico indica que o desempenho do indicador pode ser apresentado por um determinado período, exibindo os dados com valores, ações, anotações e observações. A meta é mostrada através da linha vermelha e os resultados através

das barras azuis que representam um mês de cada ano referente ao período selecionado. Este *software* implementa um BSC e mostra-se bastante completo, apresenta um grau de complexidade médio e não é uma ferramenta Web.

3.2 Softwares BSC Embutidos

Existem ferramentas para BSC que estão incorporadas em outros sistemas, principalmente sistemas ERP. Outro caso de incorporação de BSC é em sistemas de BI – *Business Intelligence*²⁸. As vantagens dessas ferramentas é a facilidade de integração com os módulos que elas proporcionam.

3.2.1 SAP

SAP é uma empresa que desenvolve soluções comerciais e as direciona para grandes, médias e pequenas empresas, mesmo assim os investimentos em recursos financeiros, humanos e de tempo de implantação são bastante altos. O SAP SBC é uma ferramenta que está embutida no SAP através de seu ERP no módulo SEM - *Strategic Enterprise Management* (Gerenciamento da Estratégia da Empresa).

A Figura 10 mostra como o BSC SAP apresenta as perspectivas adotadas na implantação desta ferramenta na Petroquímica Triunfo. Observa-se que é possível cadastrar diferentes perspectivas das sugeridas inicialmente por Kaplan e Norton (1997), neste caso chamada de “Valor Sustentável”.

²⁸ Nome dado ao processo de coleta, organização e análise de informações para a gestão de negócios.

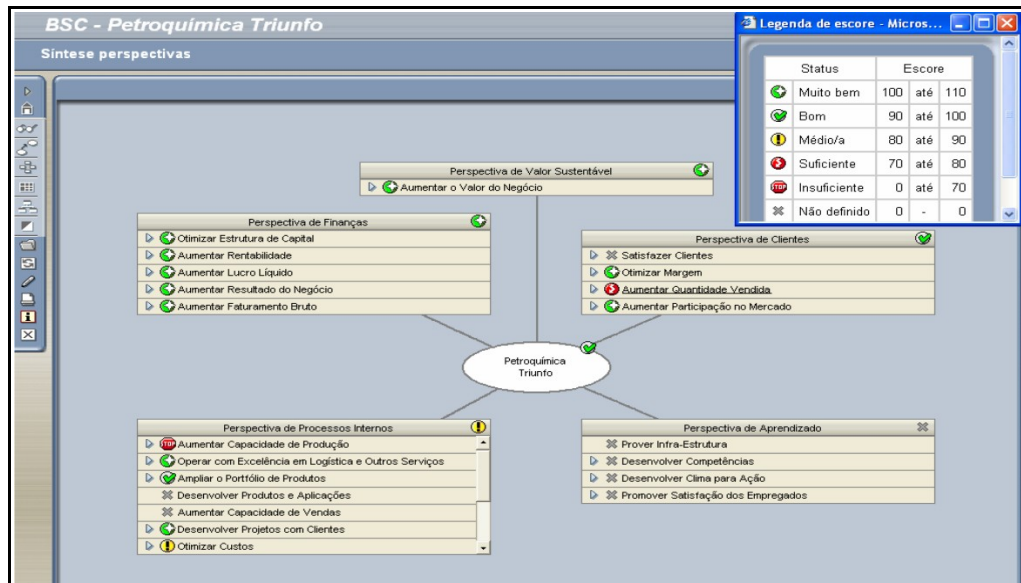


Figura 10 – Perspectivas aplicadas da Petroquímica Triunfo

Fonte: SAP (2009)

Cada uma das perspectivas vistas na Figura 10 possui seus objetivos com uma representação de como está a sua situação. Na parte superior direita desta mesma figura uma legenda representa, através de símbolos, o escore das iniciativas e perspectivas. Os símbolos representam se a situação é boa, muito boa, média, suficiente, insuficiente ou não definida. Os parâmetros são definidos no cadastramento destes indicadores e iniciativas.

3.2.2 CIGAM

O CIGAM é um ERP desenvolvido na linguagem *Magic* e possui um módulo de BSC que se baseia nas normas do BSCol, entretanto não possui essa certificação. Permite a inclusão de diversas perspectivas e mescla informações gerenciais com as estratégicas. Gera relatórios, mapas, gráficos e cubos de decisão. Não oferece interface Web (CIGAM, 2009). A Figura 11 mostra a relação de objetivos e o mapa estratégico.

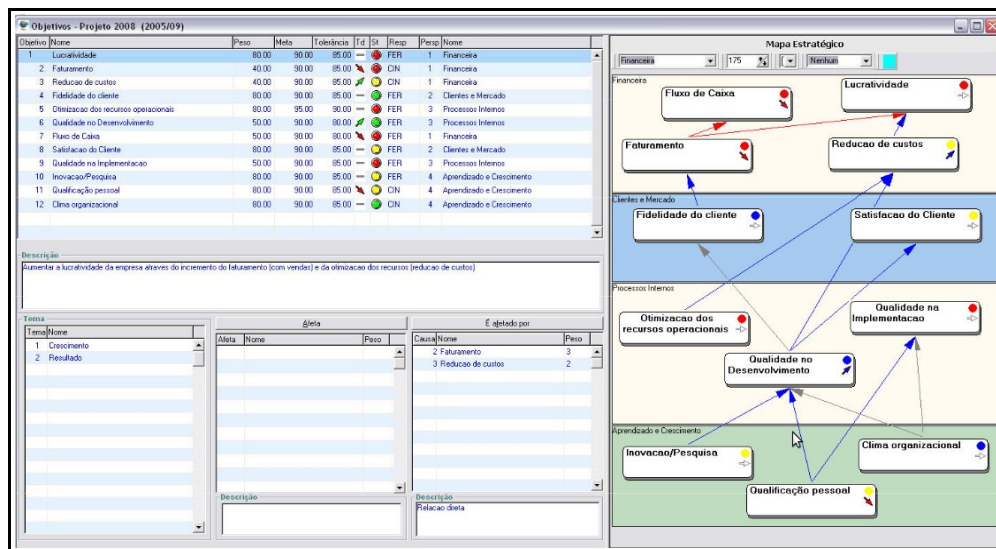


Figura 11 – Objetivos e Mapa Estratégico

Fonte: Cigam (2009)

Observa-se na Figura 11 como os objetivos se posicionam e se relacionam no mapa estratégico, indicando, tanto no mapa estratégico como na tabela a situação, que este objetivo se encontra através de um sistema sinalizador do tipo semáforo e setas que indicam o crescimento, declínio ou estagnação do desempenho do objetivo. A Figura 12 apresenta os valores dos indicadores e objetivos mostrando os gráficos de um dos indicadores, no caso, “Venda Projetos” e mostra o comportamento de um indicador em um determinado período, de acordo com os limites que lhe são atribuídos.

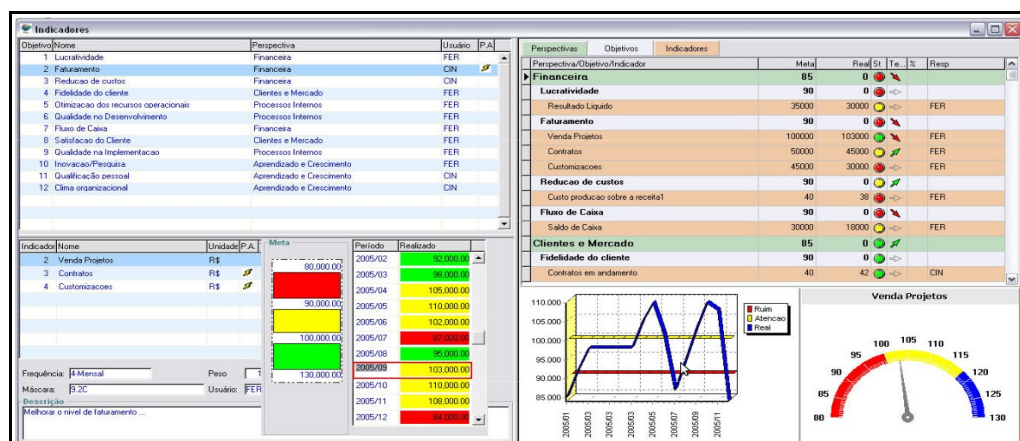


Figura 12 – Gráfico do indicador “Venda Projetos”

Fonte: Cigam (2009)

3.2.3 OpenBravo

OpenBravo (2009) é uma ferramenta Web de código aberto que integra recursos de ERP, POS – *Point of Sales* (ponto-de-venda) e BI – *Business Intelligence*. Este ambiente integrado de ferramentas administrativas oferece relatórios e análises multidimensionais através cubos OLAP²⁹. Esta análise e apresentação de resultados se dá através da integração do OpenBravo com a suíte BI, também de código aberto, Pentaho (2009). O módulo OLAP possui um painel de controle baseado em indicadores de BSC. A ferramenta é livre mas a documentação, apesar de ser vasta, não é explícita quanto a forma como implementa um BSC usando seus recursos.

3.2.4 Adempiere

Adempiere é um de ERP que surgiu em 2006 a partir do Compiere sendo esta última uma solução ERP comercial. Adempiere é distribuído sob a licença livre GPL. É desenvolvido em Java e dá suporte ao banco de dados PostgreSQL e Oracle. Disponibiliza tanto interface Web como Desktop. Semelhante ao OpenBravo, o Adempiere permite criar indicadores e acompanhar seu desempenho, entretanto, também não oferece meios de implementar um BSC de acordo com a metodologia de Kaplan e Norton. A Figura 13 mostra a tela inicial do Adempiere, que aparece logo que um usuário se autentica e entra no sistema.

²⁹ *On-line Analytical Processing* (Processo Analítico Online) é a técnica de analisar um grande volume de dados para tomada de decisão.

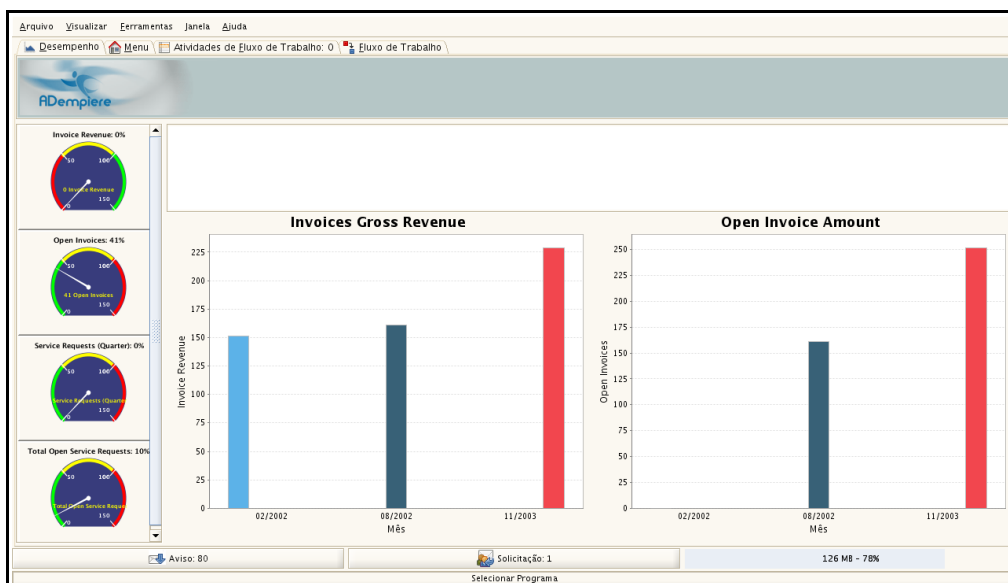


Figura 13 – Tela de abertura do Adempiere

Fonte: Adempiere (2009)

Na Figura 13 observa-se, a esquerda na vertical, o desempenho de quatro indicadores apresentados em forma de odômetro. A direita, os totais das Vendas Brutas e das Vendas em Aberto. O Adempiere usa *workflows* (fluxos de trabalho) para determinar as tarefas a serem desempenhadas. O fluxo de criação de desempenho de um indicador, inicia na escolha do esquema de cores a ser utilizado para demonstrar a performance de um indicador.

O segundo passo é preparar o cálculo que será utilizado para apurar o desempenho deste indicador. Para criar um indicador, é necessário ter um conhecimento aprofundado de como as tabelas são implementada e atualizadas no banco de dados, pois é necessário preparar a consulta, através de comandos em SQL, em determinadas tabelas, para obter os dados necessários para a obtenção dos valores que representarão este indicador. O passo seguinte é construir a medida de desempenho. Os cálculos podem ser definidos pelo usuário bem como o período de apresentação.

O último passo nesse processo é demonstrado na Figura 14, onde observa-se como o desempenho da medida “Vendas Brutas” será apresentada, qual esquema de cor será utilizado e várias outras informações como o papel que ele representa, o alvo ou meta a ser atingida, se seu escopo é anual, semestral, mensal, etc., a qual meta este indicador está subordinado, entre outros.

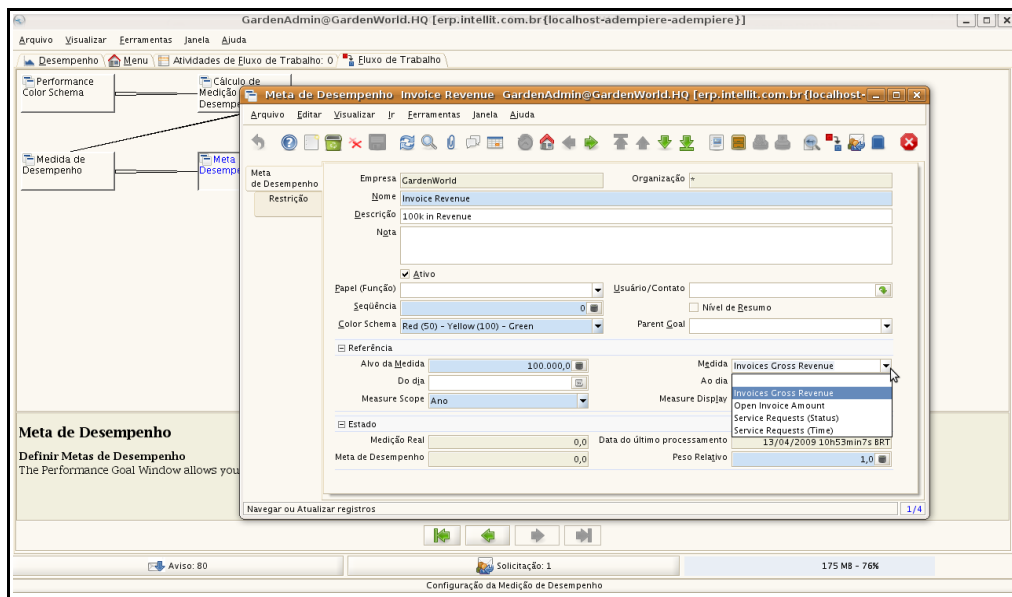


Figura 14 – Desempenho da medida “Vendas Brutas”

Fonte: Adempiere (2009)

O esquema utilizado pelo Adempiere é bastante razoável para acompanhar o desempenho de determinados indicadores, entretanto não segue a metodologia do BSC de Kaplan e Norton.

3.3 Comparativo das Ferramentas de BSC Existentes

Existe uma grande quantidade de ferramentas para BSC específicas e com BSC embutido, mas o escopo deste trabalho não é explorar exhaustivamente todas as ferramentas existentes. Foram apresentadas somente algumas para que se pudesse ter um parâmetro que fosse satisfatório e que desse subsídios para avaliar o SISCORE em comparação com as que foram explanadas na Subseção anterior.

Segundo Niven (2003) uma ferramenta de *software* para BSC deve apresentar algumas características que são consideradas fundamentais para que ela tenha uma boa classificação. Algumas das mais importantes são relacionadas no Quadro 7.

Requisito	Ferramenta							
	B S P G	O S	B O	S A	S A P	C i g a m	O B	A D
A configuração durante a instalação deve facilitada através de instruções.	N	N	N	A	A	A	A	-
Interface que gere informações de maneira ágil e amigável exibindo-as através de gráficos e painéis de controle do tipo <i>dashboard</i> (Painel de indicadores)	NI	NI	NI	A	A	A	A	A
Apresentação dos mapas estratégicos com relações de causa-efeito mostrando os valores que medem os indicadores e sinalizado-os através de cores como que em um semáforo, apresentando, também a missão e os objetivos estratégicos	A	NI	NI	A	A	A	-	-
Deve ser multi-empresa / local	A	NI	NI	-	A	-	NI	A
Proporcionar acesso ao seus dados local e remotamente, preferencialmente através Internet de forma segura	NI	NI	NI	-	A	A	A	A
Possibilitar a descrição de cada elemento do BSC, missão e visão, objetivos estratégicos, indicadores e ações, de modo a facilitar sua comunicação	A	NI	NI	A	A	A	-	-
As informações devem ser geradas, organizadas e apresentadas de formas específicas a cada tipo de usuário, desde operadores até executivos	NI	NI	NI	A	A	A	-	-
Os indicadores devem possibilitar apresentar os valores de forma consolidada, parcial e com pesos relativos	A	NI	NI	A	A	A	-	-
A apresentação dos gráficos deve ser em níveis que vão desde a empresa como um todo até o mais baixo nível de um único indicador	NI	NI	NI	A	A	A	-	-
Os indicadores que possibilitem sua apresentação de forma desdobrada ou agrupada através de níveis navegáveis intuitivamente, chamados <i>scorecards</i> em cascata	NI	NI	NI	A	A	A	-	-
Os menus de opções devem proporcionar uma navegabilidade intuitiva	NI	NI	NI	A	A	A	A	A
O sistema deve proporcionar sistemas de ajuda para todas as operação	NI	NI	NI	A	A	A	A	A
A análise dos indicadores deve permitir subir ou e descer (<i>drill-down</i>) os níveis de um indicador.	NI	NI	NI	A	A	A	-	-
O sistema deve prover análise de verificação de tendência e correlação entre os indicadores	NI	NI	NI	A	A	A	-	-
A ferramenta de BSC deve proporcionar meios de avisos aos participantes através de correio eletrônico, celular ou outros equipamentos móveis.	NI	NI	NI	A	A	A	-	-
O sistema deve oferecer compatibilidade entre os diversos sistemas operacionais existentes	A	A	A	A	A	A	A	A
As integrações com outros sistema deve ser possível	NI	NI	NI	A	A	A	NI	NI
Os resultados deve ser possíveis de serem exportados para diversos formatos	A	NI	NI	A	A	A	A	A
O sistema deve permitir o uso de diversos SGBDs	-	NI	NI	A	A	-	A	A
O controle de acesso e segurança sobre os dados deve se dar através de uma área específica para administradores do sistema	NI	NI	NI	A	A	A	A	A
Legenda: NI → não foi possível determinar, através da documentação, demonstração ou implantação, se o requisito foi atendido ou não. A → requisito atendido – a utilização do sistema ou consulta à documentação indicou que requisito foi atendido NA → requisito não atendido – a utilização do sistema ou consulta à documentação indicou que requisito não foi atendido								

Quadro 7 – Comparação das ferramentas de BSC através dos requisitos de Niven (2003)

Fonte: O autor

A maioria dos requisitos apresentados por Niven (2003) servem para quase todas as aplicações Web, mas especialmente para aplicações direcionadas a implementação de um BSC. Analisando os dados apresentados no Quadro 7, conclui-se que as ferramentas que mais atendem aos requisitos de Niven (2003), que vão de encontro com os princípios da metodologia criada por Kaplan e Norton são as específicas como SA, CIGAM e SAP. Embora

o SAP ser um ERP amplo, seu módulo SEM para BSC pode ser usado isoladamente, tornando-o uma ferramenta específica. Entretanto, observa-se a falta de iniciativas que contemplem o BSC de Kaplan e Norton através de ferramentas livres e de código aberto, apesar de existirem projetos disponíveis no SourceForge, mas inativos e com documentação insuficiente.

3.4 Casos de implementação do BSC

A aderência das empresas a uma metodologia como a de Kaplan e Norton, apesar de existir há mais de dez anos, é um processo que está em crescimento. Controle é um fator de extrema importância na saúde financeira das empresas. O BSC veio para auxiliar essa tarefa que demanda uma atenção consideravelmente grande.

Um caso sem o uso de ferramentas específicas para o BSC se deu em uma empresa de médio porte, a Werner Calçados. Outro caso é o da Grendene, uma empresa de grande porte que lançou mão de assessoria e uso de ferramentas específicas para a implementação de seu BSC.

3.4.1 O processo na Werner Calçados

Todo o histórico de como esse processo ocorreu é baseado no relato da obra de Werner Arthur Müller Júnior (2001) intitulada *Balanced Scorecard – Gestão de Informações Gerenciais para Empresa Calçadista, Metodologia de Implantação Prática*.

3.5 O início do processo

Inicialmente a administração da Werner Calçados decidiu fazer um estudo dos princípios básicos da metodologia de Kaplan e Norton. Esse estudo se fundamentou nos conceitos e objetivos da metodologia do BSC, a visão do BSC como um sistema gerencial estratégico, o estudo das perspectivas, seus indicadores e a estrutura dos sistema como um todo.

3.5.1.1 Segundo passo - Conscientização

Uma das tarefas que representam maior dificuldade é exatamente a mudança da mentalidade dos administradores e colaboradores de uma empresa que tem anos de experiência e uma forma específica de gerenciar seus negócios. Isso gera comodismo e causa desconforto e resistência quando surgem iniciativas para mudar esse processo.

Entendido todo método de como implementar um BSC, chega o momento de conscientizar a alta administração, definir os responsáveis pela implantação dessa nova forma de administrar ao negócio e fazer com que as pessoas entendam a importância dessa mudança e a relevância do envolvimento comprometido das pessoas.

3.5.1.2 Desenvolvimento

Superados os primeiros obstáculos, os administradores da Werner Calçados, em conjunto com todas as pessoas envolvidas, definiram os objetivos estratégicos, elaboraram a arquitetura dos indicadores, e definiram as metas e estratégias operacionais. Por fim, juntando todos esses elementos, foi traçado o plano de ação para a implementação do BSC.

3.5.1.3 A prática

A fundamentação para a criação dos indicadores do BSC foi bastante baseado em indicadores financeiros tais como Lucratividade, Faturamento, Vendas por Região, Índice de Liquidez, Custo Hora Médio, etc. Müller (2001) mesmo admite que coloca o BSC “*como um braço da contabilidade*”. Indicadores não financeiros como Grau de Satisfação dos Clientes, Acompanhamento do Plano de Mídia, Pontuação Conquistadas em Pesquisa de Opinião, Grau de Escolaridade dos Recursos humanos entre outros, também foram utilizados de forma sólida e com uma atenção considerável.

Os indicadores foram bastante explorados e detalhados e o princípio da relação de causa e efeito foi empregado. As perspectivas sugeridas por Kaplan e Norton foram bem aplicadas e mostraram uma abrangência coerente.

3.5.1.4 Ferramentas

A apresentação dos resultados que deve ser em níveis diferentes para cada tipo de usuário como sugere Niven (2003), fica comprometida se não for usada uma ferramenta específica para BSC e que contemple esses requisitos.

As ferramentas utilizadas para o desenvolvimento e implantação do BSC na Werner Calçados foram editores de textos e planilhas eletrônicas. Usando essa prática é possível acompanhar, atualizar e apresentar os resultados obtidos, entretanto o dinamismo e participação dos recursos humanos envolvidos no processo perde riqueza de detalhes, uma vez que não existe acompanhamento online sua operação fica restrita a poucas pessoas.

3.5.1.5 Conclusões do caso Werner Calçados

Müller (2001) conclui que os estudos auxiliaram a Werner Calçados a desenvolver um método sistêmico, direcionado especialmente à empresa de calçados, alicerçando a formulação de gestões voltadas para a estratégia e operação.

3.5.2 O caso Grendene

Este caso foi analisado através de testemunhos no site da Beringer Consulting (2009) e em participação em palestra na Sucesu (2009), onde o caso foi todo relatado pelo CIO de TI da Grendene (2009) que, apesar de também ser uma empresa direcionada à produção de calçados, relatou como foi o processo de implantação do BSC na área de TI.

Não muito diferente do caso anteriormente relatado a Grendene, além de passar por todos o processo de fundamentação, conscientização, desenvolvimento e prática, o BSC foi aplicado com o uso de um ferramenta específica para BSC.

3.5.2.1 O desafio

A ideia de poder mensurar, acompanhar e programar o desempenho e desenvolvimento na área de TI da Grendene representava um grande desafio para seu CIO Ernani Toso. Trabalhando há bastante tempo na Grendene – segundo relato, abriu a caixa do primeiro microcomputador da empresa – percebeu que o todos os serviços prestados pela sua área estavam sendo difíceis de controlar e que isso, por consequência, estava trazendo descontentamento por parte dos clientes e usuários de seus serviços.

Decidiu contratar uma consultoria na área de gerenciamento estratégico e iniciou o

processo de implementação de um BSC em seu setor. Basicamente o processo antes da implementação e uso de qualquer ferramenta foi similar ao do caso anteriormente relatado. O diferencial está na aplicação de uma ferramenta computacional direcionada para BSC.

3.5.2.2 Ferramentas

O BSC foi implementado na Grendene através do *software* para *Business Intelligence* e gerenciamento de performance financeira da IBM chamado Cognos (IBM, 2009). Esse software implementa todos os conceitos do BSC em um módulo específico para esse fim. O Cognos é um dos 25 *softwares* que é certificação pela BSCol (PALLADIUM, 2009).

3.5.2.3 Os resultados do caso Grendene

O entusiasmo do CIO da Grendene, relatando a sua experiência em resultados obtidos como uso do BSC, denotam a importância e relevância de iniciativas como essa.

“A Grendene decidiu implantar o BSC aplicado a TI para melhorar o desempenho dos produtos e serviços [...], criando e alinhando a estratégia do departamento com a estratégia do negócio Grendene. Para a gestão estratégica de TI, também foi importante definir um padrão de gerenciamento para toda equipe de comando e com isso criar um referencial para apoiar a comunicação da Informática com os demais departamentos da Grendene, além de sustentar um ambiente de trabalho orientado para a estratégia. A aplicação do BSC na Governança de TI da Grendene é muito mais do que apenas acompanhar os indicadores [...], agora acredito que podemos cumprir nossa Missão e atingir as metas, com uma equipe de maior maturidade profissional [...] superou todas as expectativas (TOSO, Ernani. Beringer Consulting, Disponível em <<http://www.beringer.com.br>>. Acesso em: 11 Mar 2009).

Diante dos relatos de Toso e de Müller, pode-se mensurar a importância que uma ferramenta administrativa como o BSC significa para quem a aplica, principalmente com o uso de um *software* que contemple os requisitos sugeridos por Niven (2003).

4 SISCORE: UMA FERRAMENTA WEB PARA BSC

O SISCORE é um *software* que implementa os principais requisitos do BSC. Esses requisitos são atendidos através de uma aplicação Web que possibilita aos seus usuários manterem acesso remoto, através da Internet ou intranet, e a todas as informações segundo as permissões contidas no banco de dados.

4.1 Descrição do SISCORE

Uma ferramenta como o SISCORE usa vários cadastros que compreendem dados gerais como pessoas, usuários, etc., e dados específicos ao BSC que são as estratégias, perspectivas, planos de ação, indicadores, etc. Todos os cadastros são necessários para que se possa implantar um BSC e torná-lo funcional.

Aos usuários do SISCORE foram previstas permissões específicas estabelecendo os limites de cada um a fim de aumentar o controle e evitar problemas de segurança, facilitando o acesso somente às informações pertinentes ao usuário.

Uma vez inseridos os dados dos cadastros essenciais, definidos os papéis de cada usuário e atribuídas as permissões, é possível estabelecer metas para os indicadores, correlacioná-los e atribuir a responsabilidade pela sua movimentação. Esses usuários deverão observar as metas e prazos prefixados pelos administradores do BSC.

Os resultados são apresentados através de gráficos e mapas que possibilitam ao administrador efetuar o acompanhamento através dos apontamentos do desempenho dos indicadores e das perspectivas que a eles estão associadas.

4.2 Desenvolvimento do SISCORE

O desenvolvimento do SISCORE deu-se por meio de recursos tais como ferramentas e metodologias aplicadas a engenharia de *software* e diversas linguagens de programação. O processo de *software* adotado foi o sequencial linear, pois as funcionalidades, os conceitos e os resultados esperados de um *softwares* de BSC são conhecidos. Esses elementos determinam que esse tipo de processo mostra-se o mais adequado. Diferentemente de processos de *software* incrementais em que as funcionalidades são definidas juntamente com o cliente e de forma iterativa, ou seja, em repetições de um determinado ciclo de processo.

Todas das ferramentas usadas para o desenvolvimento do SISCORE são de código aberto. A principal linguagem de programação utilizada para o desenvolvimento foi PHP 5 seguida de Javascript, HTML e CSS. A programação usou o paradigma de Programação Orientado a Objeto (POO) com o Padrão de Projeto *Model View Controller* (MVC). Foi utilizado o *framework* Yii, considerando que é um dos mais recentes e que melhor se adaptou aos padrões propostos nesse projeto. As informações são armazenadas em banco de dados independente de plataforma. A ferramenta IDE - *Integrated Development Environment* (Ambiente Integrado de Desenvolvimento) usada para escrever os programas foi a NetBeans³⁰.

4.2.1 Análise

A fase de análise em um projeto é um item de extrema importância, considerando que representa a definição das funcionalidades do sistema. Neste momento deve-se dispender esforços suficientes para que ela possa prover uma documentação rica para seu desenvolvimento.

³⁰ Ferramenta que proporciona ambientes integrados para o desenvolvimento de *software* desenvolvida na linguagem de programação Java (<http://www.netbeans.org>)

4.2.1.1 Requisitos Funcionais

A relação de requisitos funcionais foi criada baseado-se na proposta de Niven (2003). Esses requisitos representam “o quê” o sistema deve fazer do ponto de vista de suas funcionalidades. A relação destes requisitos com os Casos de Uso é quase direta SCOTT (2003). No Quadro 8 estão relacionados e descritos todos os requisitos funcionais relativos ao SISCORE.

Requisito Funcional	Descrição
1 – Mapa estratégico	O sistema deve apresentar mapas estratégicos com a respectivas relações de causa e efeito mostrando seus indicadores de forma que estejam sinalizando seus estados através de cores.
2 – Objetivos	O sistema deve apresentar junto aos mapa estratégico seus objetivos com seus indicadores, metas e iniciativas (ações / tarefas).
3 – Multi-empresa / Local	O sistema deve permitir que se cadastre várias empresas e vários locais relacionados às empresas.
4 – Informações personalizadas por grupo de usuário	O sistema deve gerar, organizar e apresentar as informações de forma específica a cada grupo de usuário, de acordo com suas permissões.
5 – Ajuda	O sistema deve prover ajuda para todas as operações.
6 – Avisos	O sistema deve proporcionar que os participantes recebam avisos por correio eletrônico e em equipamentos móveis.
7 – Resultados Formatados	Os resultados, gráficos e relatórios, devem ser possíveis de serem exportados os formatos mais utilizados.
9 – Usuários	O Sistema deve prover um Cadastro de Usuários do Sistema onde devem conter dados de identificação como nome, contato e as permissões que ele terá sobre os cadastros, módulos e funcionalidade do sistema, permitindo registrar o tipo de usuário, administrador, operador, etc.
10 – Estratégia	O Sistema deve prover um Cadastro de Estratégia que descreva a visão, missão, objetivos estratégicos, indicadores e ações, bem como indicar quais perspectivas a ela se relacionam.
11 – Perspectivas	O Sistema deve prover um Cadastro de Perspectivas, onde seja possível informar sua descrição e correlacionar os Objetivos, Indicadores, Metas e Iniciativa, de acordo com o modelo de Kaplan e Norton. Também deve indicar como elas se relacionam com outras perspectivas.
12 - Indicadores / Medidas de Desempenho	O Sistema deve prover um Cadastro de Indicadores / Medidas de Desempenho onde serão apontados uma descrição, sua meta, os Produtos/Serviços a que ele se relaciona, a unidade de medida que ele representa, os níveis / limites de sinalização (acima, dentro, fora) dos valores a ele atribuídos, bem como as iniciativas / ações que devem ser tomadas de acordo com seus níveis. Deve ainda possuir uma entrada de dados para os usuários, de acordo com suas permissões, informar os índices e níveis atingidos. O sistema deve permitir a apresentação dos indicadores de forma consolidada, parcial, com pesos relativos, desdobrada ou agrupada e através de níveis com navegação intuitiva e em cascata (<i>drill-down</i>).
13 – Correlação entre os indicadores	O Sistema deve permitir ao usuário correlacionar os indicadores de forma que o sistema mostre qual indicador está interferindo positiva ou negativamente.
14 – Tabela Semáforo	O Sistema deve apresentar, quando solicitado, uma tabela relacionando os indicadores sinalizando os níveis atingidos, filtrando-os por período, perspectiva e níveis atingidos.
15 – Gráficos de Desempenho	O Sistema deve apresentar, quando solicitado, gráficos de desempenho dos indicadores sinalizando os níveis atingidos, filtrando-os por período, perspectiva e nível atingido. A apresentação deve ser em níveis que vão desde a empresa até um único indicador.

Quadro 8 – Requisitos Funcionais

Fonte: o autor

4.2.1.2 Requisitos Não Funcionais

Os requisitos não funcionais estão relacionados a qualidades tais como desempenho, segurança, confiabilidade, escalabilidades e usabilidade. SCOTT (2003). No Quadro 9 estão descritos os requisitos não funcionais especificados para o SISCORE.

Requisito Não Funcional	Descrição
1 – Configuração	O sistema deve oferecer documentação com instruções detalhadas que facilitem a instalação e configuração do <i>software</i> .
2 – Apresentação	A forma de apresentação das telas e gráficos deve ser amigável (<i>friendly interface</i>) e através de painéis de controle que possibilitem uma visualização intuitiva e funcional.
3 – Segurança	O sistema deve estar instalado em servidores que disponham de todos dispositivos de segurança necessários para evitar invasões.
4 – Compatibilidade	O sistema deve ser compatível entre os diversos sistemas operacionais existentes. Essa compatibilidade deve ocorrer sem que se altere os programas fontes. As alterações devem ficar restritas à arquivos de configuração.
5 – Apresentação diferenciada	O sistema deve ter padrões de apresentação dos dados para os diferentes tipos de usuários. Por exemplo, quando um usuário com perfil de administrador, que possui características próprias identificar-se no sistema, este deve apresentar as opções de acordo com seu perfil.
6 – Navegabilidade	A navegabilidade dos menus deve ser intuitiva, possibilitando ao usuário chegar facilmente ao local pretendido.
7 – Acesso Seguro	O sistema deve proporcionar acesso local e remoto de forma segura. O acesso remoto deve ocorrer através dos principais navegadores existentes.
8 – Integração	O sistema deve prover meios para integração com outros sistemas, tanto em aplicações Web como Desktop.
9 – Independência de banco de dados	O sistema deve permitir o uso dos principais bancos de dados existentes.

Quadro 9 – Requisitos Não Funcionais

Fonte: o autor

4.2.2 Projeto

O SISCORE foi desenvolvido usando os principais fundamentos do Processo Unificado. Essa prática permite que se desenvolva uma documentação de fácil leitura no que se refere a interpretação do sistema e de suas funcionalidades como um todo. Nas próximas seções serão apresentados e detalhados os diagramas de caso de uso, os diagramas de classes e o diagrama E-R. Os demais diagramas, também bastante utilizados no processo unificado, como o de sequencia e o de atividades, não foram implementados porque o SISCORE é um

sistema que apresenta poucos ou, em muitos casos de uso, nenhum fluxo alternativo, baixa complexidade e suas funcionalidades são conhecidas.

4.2.2.1 Casos de Uso

Um caso de uso representa uma sequência de ações executadas em um sistema por um ator. Um diagrama de casos de uso representa “o que” o sistema deve fazer, mas não diz “como” deve ser feito.

O caso de uso representa os atores que efetuam interações com o sistema. Um ator dentro de um sistema pode representar um papel que um usuário desempenhe ou uma entidade externa como um outro sistema interagindo com o sistema em questão (SCOTT, 2003).

Os atores identificados para o SISCORE são Administrador, Operador, Responsável, Sistema e Analista de Negócio. O Administrador é o responsável pelas tarefas que demandam a responsabilidade pela segurança do sistema e o Operador é responsável pelos cadastramentos e outras inclusões de dados, o ator Responsável alimenta os dados dos indicadores, o Sistema avisa os Responsáveis quanto os prazos saírem do planejado e o Analisa de Negócio avalia os resultados alcançados. As Figuras 15, 16, 17, 18 e 19 ilustram os casos de uso conforme os atores supra citados.

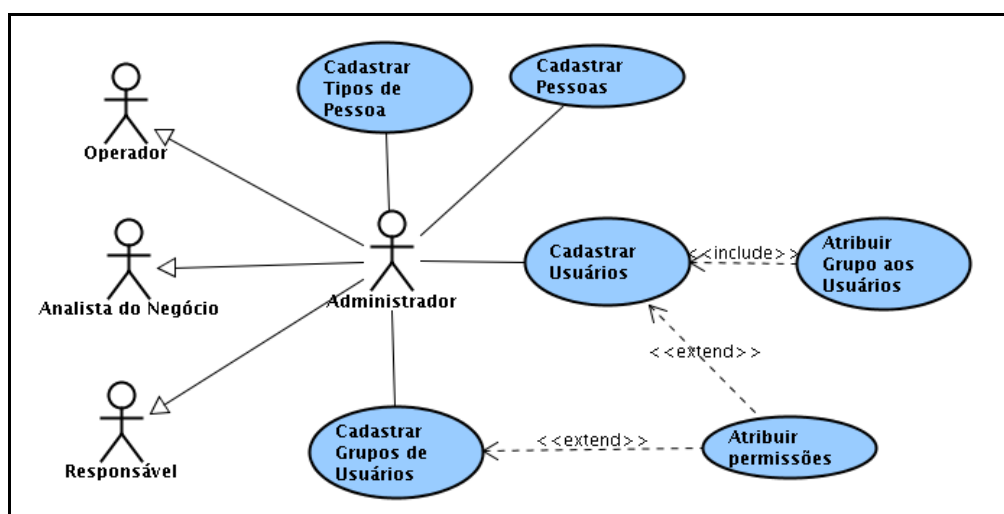


Figura 15 - Casos de uso Administração

Fonte: o autor

O caso de uso ilustrado na Figura 15 mostra quais são as atividades que um usuário desempenha no papel de administrador do sistema. Um usuário administrador herda todas as atividades desempenhas pelo usuário Operador, pelo usuário Analista do negócio e pelo usuário Responsável. O diagrama da figura 16 apresenta os casos de uso atribuídos ao ator Operador. Os casos de uso foram categorizados, diferenciando-os através das cores.

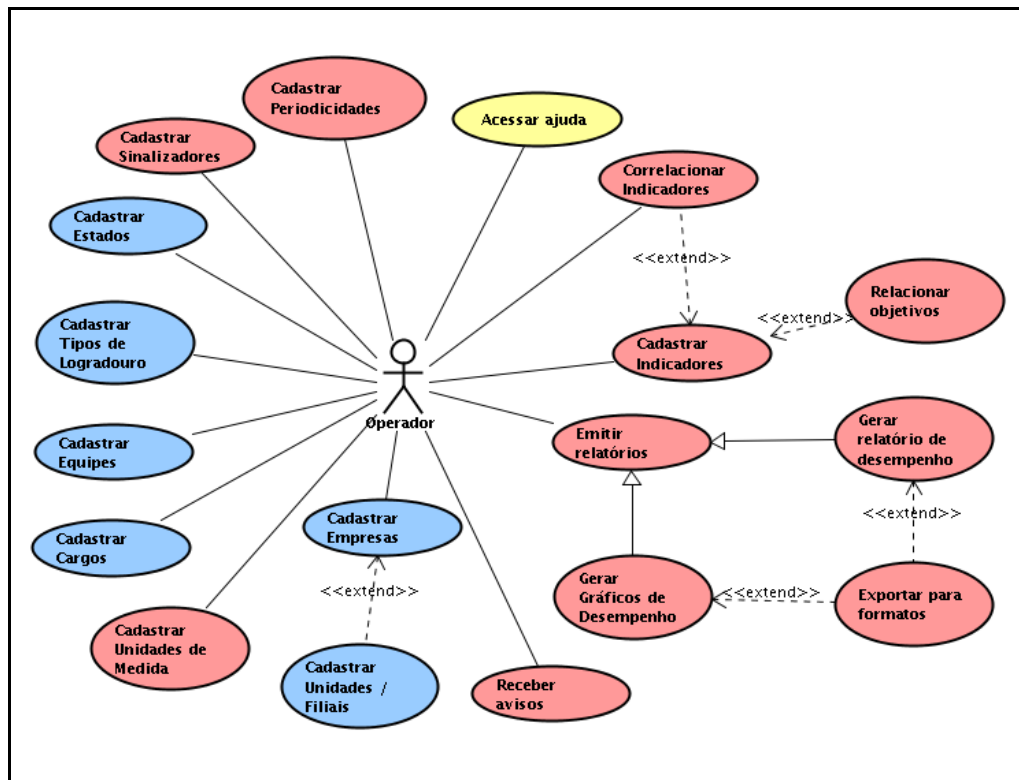


Figura 16 - Casos de uso Operação

Fonte: o autor

As cores que diferenciam os casos de uso da Figura 16. Azul representa as funcionalidades e cadastros gerais, que são comuns às diversas aplicações; as funcionalidades e cadastros específicos do BSC estão representadas pelo vermelho, e as demais, típicas a qualquer sistema, em amarelo. Esse critério foi aplicado em todos os casos de uso para facilitar a separação e a visualização das especificidades. Na Figura 17 estão representados os casos de uso do ator Responsável.

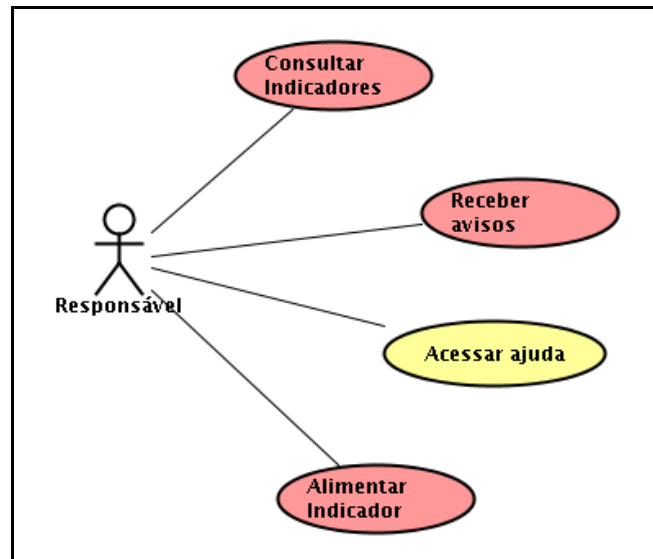


Figura 17 - Casos de uso entrada de dados

Fonte: o autor

A Figura 18 apresenta o único caso de uso do ator Sistema. Frequentemente, mas não sempre, os atores são representados por pessoas mas, como no caso de uso Sistema, existem atores que não são humanos mas interagem com casos de uso.

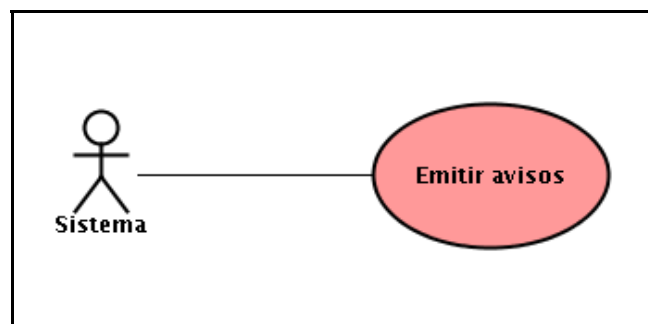


Figura 18 - Casos de uso Sistema

Fonte: o autor

A ação de “Emitir Avisos” é um exemplo de caso de uso cujo processo é disparado pelo próprio sistema a partir de parâmetros previamente informados ao SISCORE que automatizam essas operações. O diagrama da Figura 19 mostra os casos de uso relacionados ao ator Analista do Negócio.

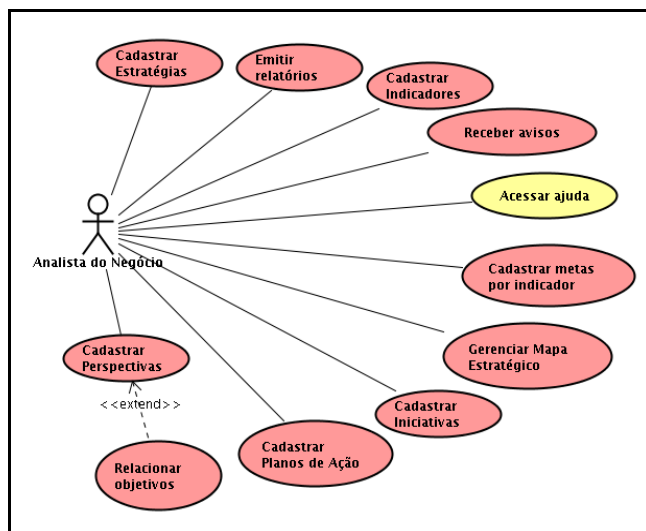


Figura 19 - Casos de uso Análise do Negócio

Fonte: o autor

O Analista do Negócio, ilustrado na figura 19, tem a função de concentrar-se no negócio que, por sua vez, deve tratar dos casos de uso mais diretamente relacionados a metodologia BSC.

4.2.2.2 Detalhamento dos casos de uso

O diagrama caso de uso é um recurso usado para criar “[...] um conjunto de cenários que identifica uma linha de uso para o sistema a ser construído.” (PRESSMAN, 2002, p. 273). Portanto, faz-se necessário um detalhamento de cada ação executada pelos respectivos atores, explicitando como as ações serão executadas.

Um detalhamento de caso de uso é uma tarefa que “[...] envolve escrever o texto para o curso básico e para cursos alternativos de ação [...] para um caso de uso” (SCOTT, 2003, p. 45).

No Apêndice A é apresentado o detalhamento do caso de uso Alimentar Indicadores. A proposta é criar um padrão a partir de um modelo de fácil visualização e aplicável a qualquer caso de uso, diferente do proposto pelo processo unificado que é apenas um texto que “[...] deve geralmente conter de um a três parágrafos de extensão; o texto para cada curso alternativo deve ter em média uma sentença ou duas” (SCOTT, 2003, p. 45).

4.2.2.3 Diagramas de Classe

Segundo Pressmann (2002, p. 536) uma classe “[...] é um conceito de OO que encapsula as abstrações de dados e procedimental necessárias para descrever o conteúdo e o comportamento de alguma entidade do mundo real” (PRESSMAN, 2002, p. 536).

O *framework* Yii, através de ORM, gerou grande parte das classes do SISCORE. Poucas classes, mais relacionadas às regras de negócio, é que foram criadas posteriormente a essa geração automática. O diagrama da Figura 20 apresenta as classes de um CRUD³¹ no SISCORE e como elas se relacionam com as classes nativas do Yii.

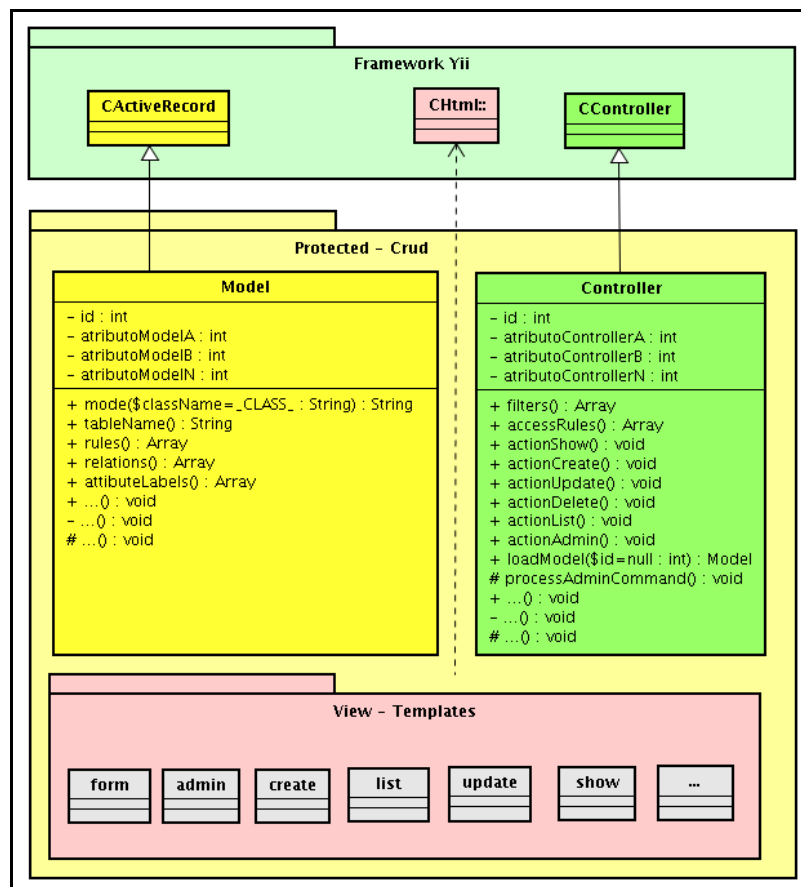


Figura 20 – Estrutura das classes de um CRUD no SISCORE

Fonte: o autor

Antes das classes serem geradas com ORM do Yii, o esqueleto da aplicação é gerado com um *script* (ver Subseção 4.2.3.2). Essa geração reserva uma área destinada

³¹ Acrônimo para *Create* (criar), *Retrieve* (recuperar), *Update* (atualizar) e *Delete* (excluir). Usado para definir as quatro operações básicas em um bancos de dados.

especificamente para os arquivos que contém os programas que implementam o MVC, chamada “*protected*”. Nesta área estão as classes “*Model*” e “*Controller*” e os arquivos “*View - Templates*”. Na Figura 20 estão representadas as classes onde estão contidos os atributos (características) e métodos (comportamentos) que são comuns a todas as classes.

Todas as classes “*Model*” são especializações ou generalizações da classe “*CactiveRecord*” e possuem seus próprios atributos, exemplificados nesta classe como “*id*”, “*atributoModelA*”, “*atributoModelB*” e “*atributoModelN*”. As classes “*Model*” são responsáveis por efetuar todas as tarefas relacionadas à interações da aplicação com o banco de dados. Essas classes também são compostas por métodos padrão para todo CRUD que o Yii gera automaticamente, quais sejam, “*model*”, “*tableName*”, “*rules*”, “*relations*” e “*attributeLabels*”. Esses métodos podem ser alterados (adaptados às necessidades da aplicação). Também é possível acrescentar novos métodos à classe no lugar dos métodos representados na classe “*Model*” como “*+ ...()void*”, “*- ...()void*” e “*# ...()void*”.

As classes “*Controller*”, que são uma especialização da classe “*CController*”, apresentam o mesmo esquema de atributos mas possuem outros métodos que, assim como na classe “*Model*” também podem ser alterados e acrescidos de outros métodos.

A área que na Figura 20 é denominada “*View – Templates*” compõe os arquivos que não são classes e sim *templates*, chamadas assim em um padrão MVC por serem responsáveis por possuir os modelos que a aplicação deve obedecer para exibir a visualização dos dados. Esses arquivos, por não serem classes, não são generalização de nenhuma classe do Yii, mas usam a classe “*Chtml*” que contém os atributos e métodos específicos para a apresentação de dados ao usuário. Os diagramas de classes ilustrados nas Figura 21 e 22 são essencialmente classes do tipo “*Model*” e apresentam os atributos específicos de cada classe.

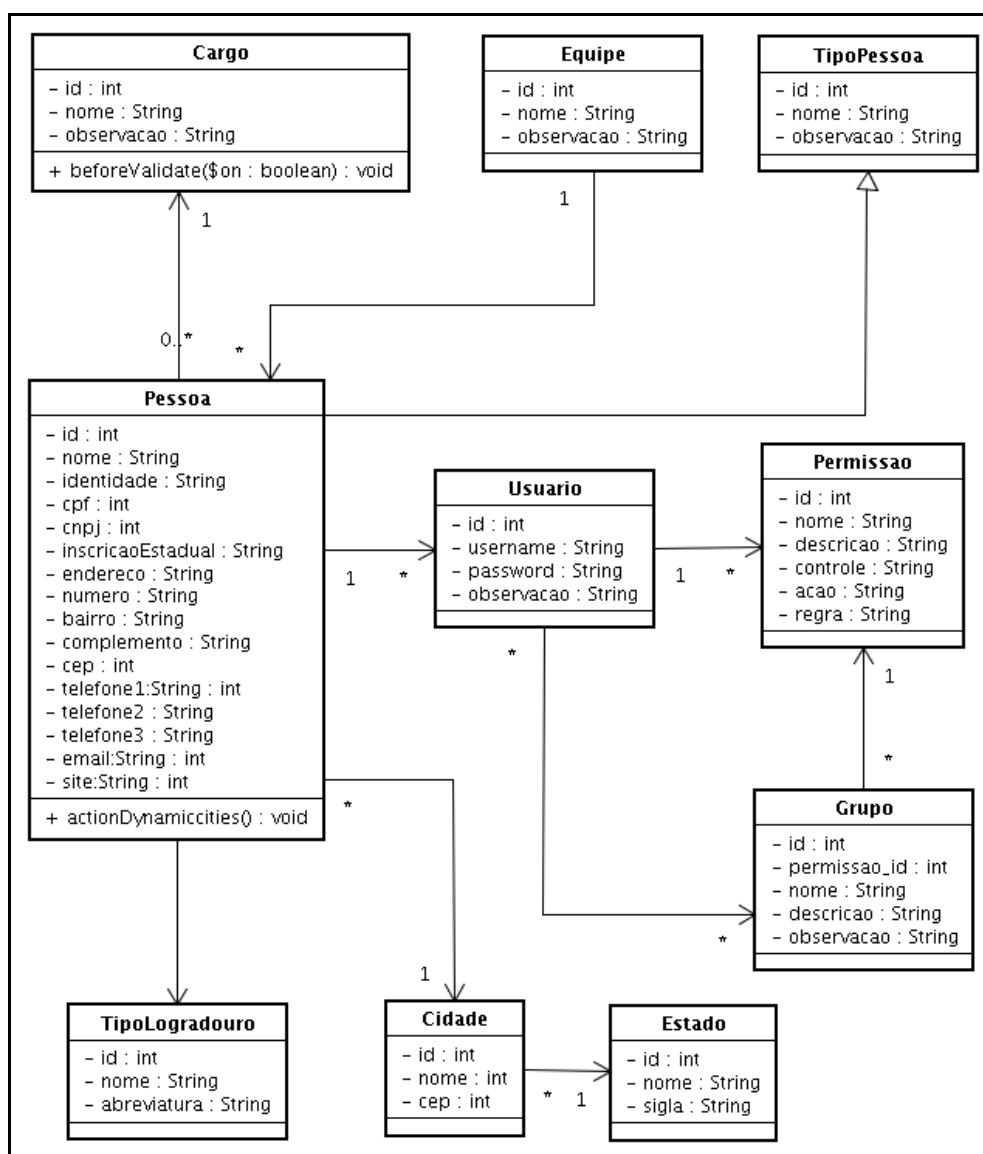


Figura 21 – Diagrama de classes – cadastros gerais

Fonte: o autor

A ilustração da Figura 21 contém as classes que relacionam dados gerais comuns a maioria das aplicações comerciais, tais como pessoas, usuários, grupos, entre outros. Os métodos que são diferentes às outras classes são explicitados no diagrama, como a classe “Pessoa” que possui o método “actionDynamiccities”, que executa uma ação específica para a classe “Pessoa”. As demais classes, onde não aparecem métodos relacionados, são comuns a classes do tipo “Model”. Esses métodos são representados na Figura 22, onde são ilustradas as classes mais específicas ao negócio BSC como indicador, estratégia, plano de ação, perspectiva, entre outros.

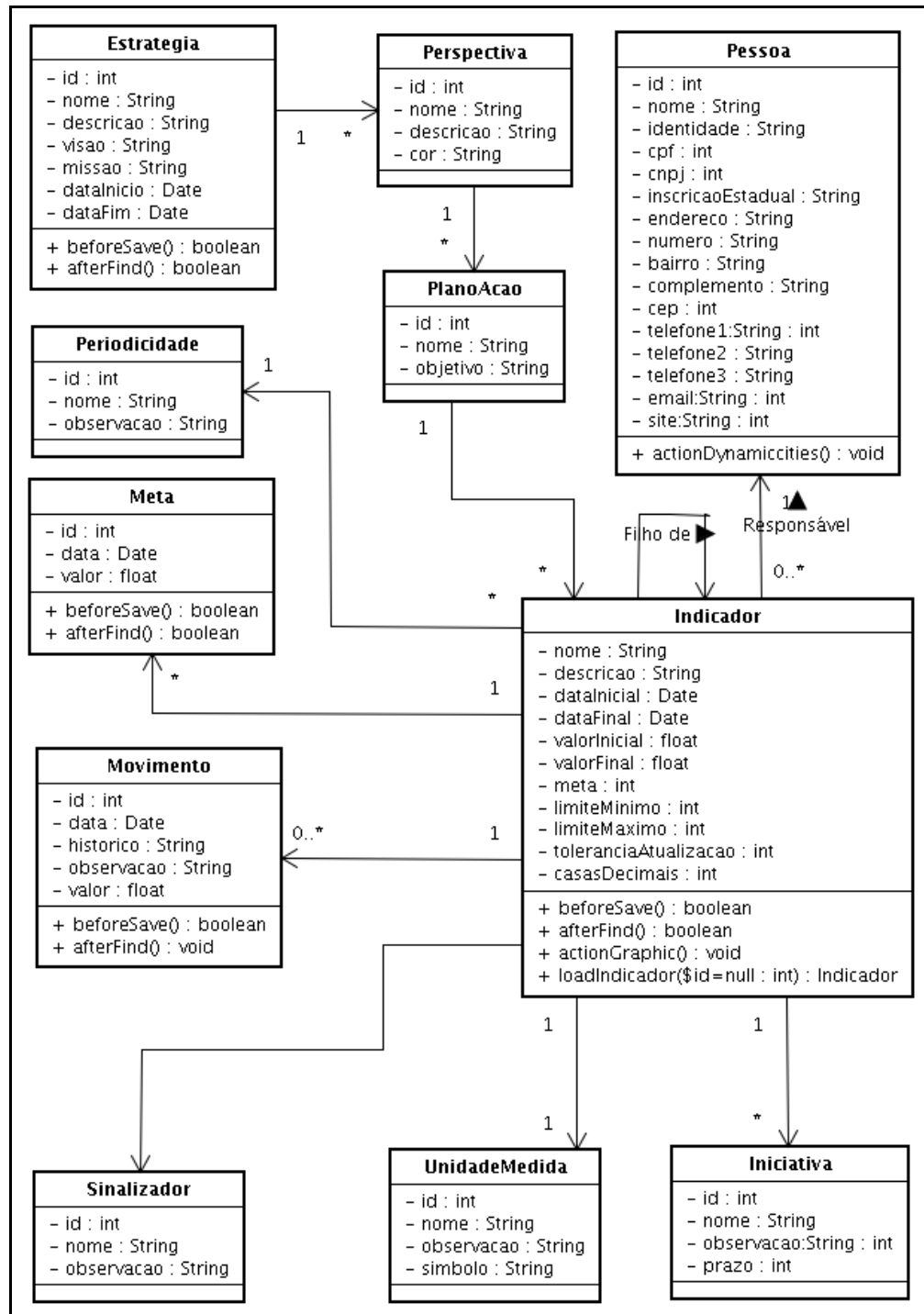


Figura 22 – Diagrama de classes – cadastros específicos

Fonte: o autor

No diagrama da Figura 22 aparece outro elemento: a classe “Pessoa” que explicita o tipo de relacionamento que existe entre ela e a classe Indicador, ou seja, todo indicador possui um responsável que é uma classe do tipo uma Pessoa. É importante ressaltar que no diagrama de classes da Figura 22 existem mais métodos específicos a determinadas classes como na classe “Indicador” cujos métodos são “beforeSave”, “afterFind”, “actionGraphic”,

”loadIndicador”. Esse fato denota a especificidade dessas classes que estão mais relacionadas às regras de negócio do BSC do que a regras elementares dos cadastros comuns a maioria das aplicações.

4.2.2.4 Projeto E-R

Um projeto E-R (Entidade-Relacionamento) é um modelo conceitual para definir as tabelas do banco de dados e suas relações. Para a modelagem e criação do banco de dados do SISCORE foi usada a ferramenta DBDesigner Fork³². Assim como nos diagramas de classes, tendo em vista o mapeamento 1 para 1, os diagramas E-R também foram separados em dois: específicos e gerais. Semelhante ao diagrama de classes, que mostra os relacionamentos, atributos e métodos, o diagrama E-R apresenta somente o relacionamento entre as tabelas (representadas como entidades), os atributos e seus respectivos tipos.

A Figura 23 e 24 apresentam os diagramas E-R que deram origem ao banco de dados e as tabelas do SISCORE. O outro diagrama E-R, ilustrado na Figura 24, mostra as tabelas especificamente relacionadas a um *Balanced Scorecard*.

³² Ferramenta visual que integra o projeto de relacionamento de entidades, de criação de banco de dados e gera *scripts* SQL para a criação de bancos de dados Oracle, SQL Server, MySQL e Firebird (<http://sourceforge.net/projects/dbdesigner-fork/>)

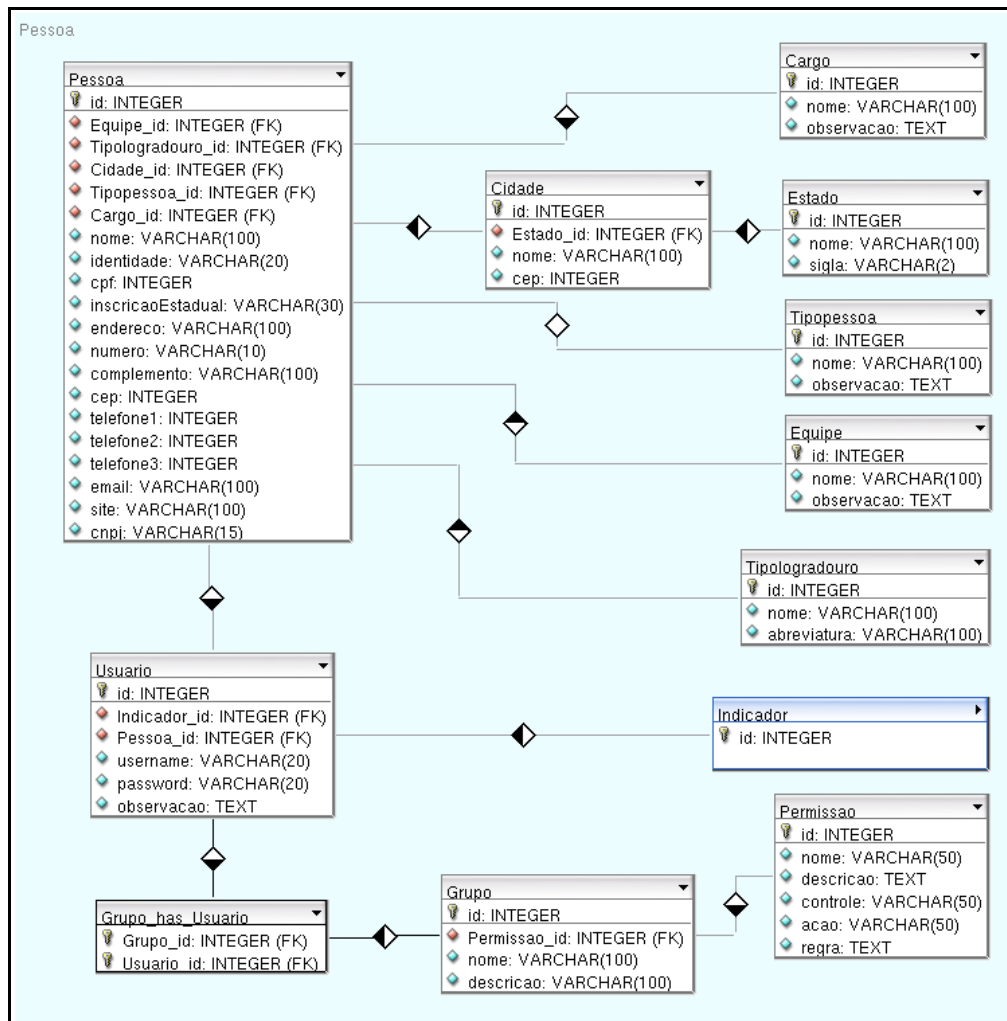


Figura 23 – Diagrama E-R – cadastros gerais

Fonte: o autor

No diagrama da figura 24 existe um relacionamento da tabela “Indicador” com ela mesma, que representa um caso de hierarquia de indicadores, onde um indicador pode ser pai ou filho de outro indicador. O mesmo caso de hierarquia é representado também no diagrama de classes da Figura 22.

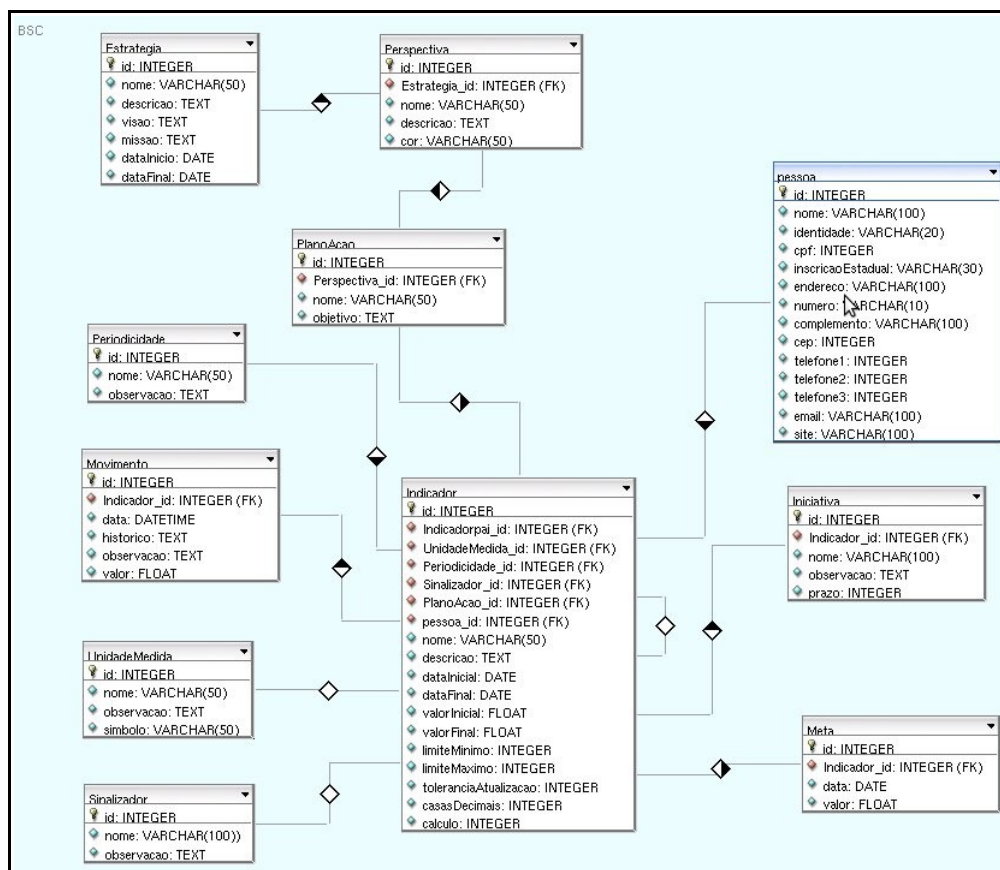


Figura 24 – Diagrama E-R – cadastros específicos

Fonte: o autor

4.2.3 Codificação

Na codificação do SISCORE foi usando o paradigma de Programação Orientado a Objeto (POO) e aplicado o padrão de projeto Modelo-Visão-Controle (MVC). A implementação deste padrão de projeto foi feita com o uso do *framework* Yii. O uso de uma ferramenta IDE como o NetBeans, também auxiliou no desenvolvimento, pois dispõe, como na maioria das principais e melhores ferramentas IDEs, vários recursos que facilitam a programação.

4.2.3.1 Estrutura da aplicação

A estrutura do SISCORE está intimamente ligada à proposta do Yii. Aplicando o padrão de projeto MVC foi obtido um *software* cuja estrutura está representada pelo diagrama da Figura 22.

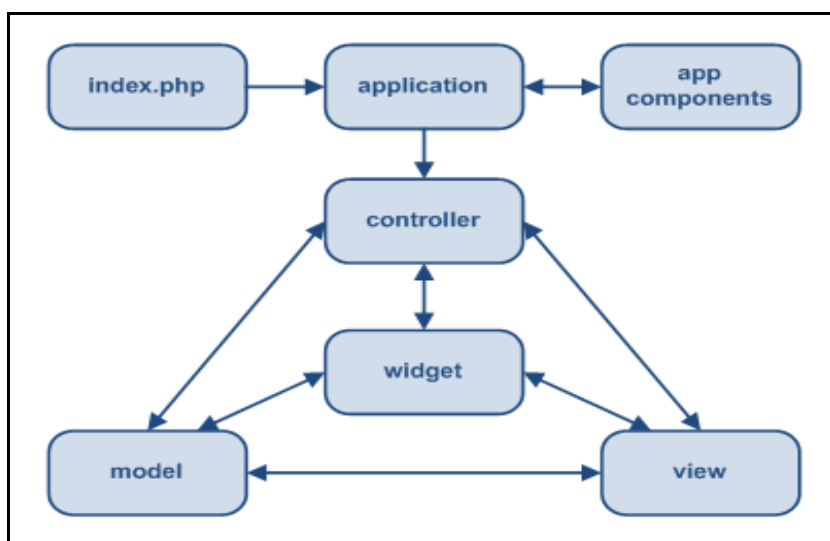


Figura 25 – Estrutura de uma aplicação com Yii

Fonte: Yii (2009)

Pode-se observar no diagrama da Figura 21 que o Yii usa o padrão de projeto *front controller* (controle frontal) que “consolida todo tratamento de solicitações canalizando-os através de um único objeto manipulador” (FOWLER, 2006, p. 328). No caso do Yii o *front controller*, chamado de aplicativo, recebe as solicitações do usuário através do Index e as envia para o controlador para que possam ser devidamente processadas. Para entender melhor como o SISCORE é executado, na Figura 26 está representado o fluxo das aplicações implementadas através do uso do *framework* Yii.

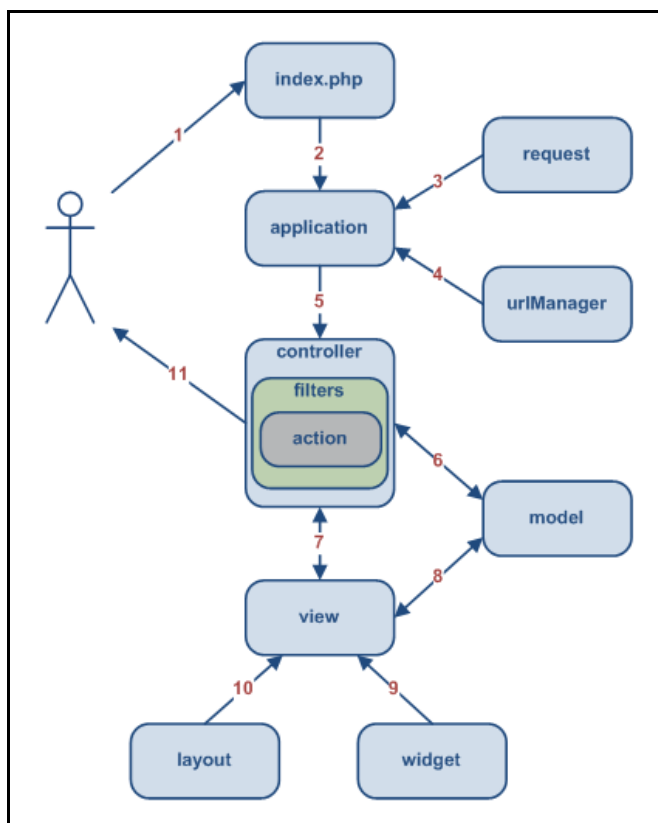


Figura 26 – Fluxo de execução do SISCORE

Fonte: Yii (2009)

Os passos do fluxo da Figura 26 representam a execução do SISCORE e são descritos como segue:

1. quando o usuário faz uma solicitação com a URL³³, como `http://localhost/SISCORE/index.php?r=indicador/show&id=1`, o servidor Web processa o pedido executando o *script* de *bootstrap*³⁴ “index.php”;
2. neste momento é criada e executada uma instância³⁵ da classe denominada “application”;
3. a partir deste momento “application” obtém as informações da solicitação através do componente chamado “request”;
4. a “application” determina o controle e a ação solicitada através do componente “urlManager”. Neste exemplo o controle é “indicador” referindo-se à classe “indicadorController”. A “action” “show” é determinada no “controller”;
5. a “application” cria uma instância do “controller” para tratar a solicitação do usuário. O “controller” determina que a ação “show” refere-se a um método

³³ Endereço de um recurso localizado na rede, seja Internet ou intranet.

³⁴ Arranque, carregamento, inicialização.

³⁵ Concretização, criação de uma classe em sistemas orientados a objeto.

chamado “actionShow” no “controller” da classe. Em seguida, cria e executa “filters” como o controle de acesso que são associados a esta ação que, por sua vez só é executada se passar pelos filtros;

6. a ação lê um “model” “Indicador” com ID 1 (código de identificação identificação no Banco de Dados);
7. a “action” processa a “view” chamada “show” do “Indicador” ;
8. a “view” apresenta os atributos / dados do “model” “Indicador”;
9. a visão executa alguns “widgets”³⁶;
10. a resultado do processamento da “view” é embutido em um “layout”;
11. a “action” conclui o processamento da “view” e exibe o resultado ao usuário.

Na execução do fluxo ilustrado na Figura 26 são aplicadas técnicas como *lazy loading* (carregamento tardio) que tem por princípio não incluir um arquivo de classe até que ela seja usada pela primeira vez, e não criar um objeto até que ele seja acessado pela primeira vez. Isso melhora consideravelmente o desempenho.

Entretanto, em alguns casos, essa técnica tem de ser substituída por exatamente o oposto, através da técnica de *eager loading* (carregamento ansioso). Isso é aplicado em cenários onde é necessário acessar todas as informações de um determinado registro (Yii, 2009). Por exemplo, retornar, através de uma consulta, todos os indicadores sobre os quais um usuário é responsável.

4.2.3.2 Estrutura de diretórios

O SISCORE segue a estrutura de diretórios criada automaticamente pelo *framework* Yii. Essa estrutura é apresentada na Figura 27. No diretório *protected* encontram-se os códigos desenvolvidos especialmente para o SISCORE, mais especificamente nos subdiretórios *controllers*, *models* e *views*.

³⁶ Componente gráfico aplicado aos dados do “model” durante a apresentação da “view”.

assets/	Arquivos de recursos publicáveis
css/	Arquivos CSS
images/	Imagens da aplicação
themes/	Temas de apresentação
protected/	Arquivos protegidos
components/	Componentes reutilizáveis
config/	Arquivos de configuração
controllers/	Contém as classes controle
extensions/	Extensões de terceiros
models/	Classes do modelo
runtime/	Arquivos de log
views/	Visões

Figura 27 – Estrutura de diretórios do SISCORE

Fonte: o autor

Essa estrutura pode ser alterada e personalizada, entretanto isso implica em ter um trabalho maior no momento em que houver a necessidade de atualizar uma versão do Yii, ou de qualquer componente a ele associado.

As atualizações são importantes, principalmente em ferramentas usadas para implementação de aplicações Web. É normal que a comunidade de desenvolvedores encontre problemas que devem ser corrigidos, principalmente quando esses problemas são de vulnerabilidades quanto a segurança da aplicação ou dos dados armazenados por ela.

4.2.4 Testes

Para os testes do SISCORE, foi aplicado o método denominado “caixa preta” também conhecido como teste comportamental. Esse tipo de teste permite exercitar todo o conjunto de requisitos funcionais e busca encontrar erros em funções, interfaces, estrutura de dados, erros de comportamento, de inicialização e de encerramento.

A forma de aplicação desses testes foi a não sistemática, observando o reflexo das ações no banco de dados portanto, a atenção do teste de caixa preta é focado no domínio da informação (PRESSMAN, 2002).

5 UTILIZAÇÃO DO SISCORE

O SISCORE é um *software* voltado especificamente para a implementação de um BSC. Seu escopo é definido pela metodologia desenvolvida por Kaplan e Norton e baseado nos requisitos de Niven (2003). A metodologia e os requisitos aplicados ao SISCORE buscam proporcionar um ambiente de fácil utilização e recursos com características de uso intuitivo.

Foram usados, na aplicação do SISCORE, índices e padrões fornecidos pelo MEC – Ministério da Educação através do SINAES – Sistema Nacional de Avaliação do Ensino Superior, constantes no Manual de Orientações Gerais para o Roteiro da Auto-avaliação das Instituições (INEP, 2009).

5.1 Padrões de avaliação SINAES

Os padrões definidos pelo SINAES – Sistema Nacional de Avaliação da Educação Superior sugerem e abordam dimensões e indicadores específicos com o objetivo de identificar as potencialidades e insuficiências dos cursos e instituições com o intuito de promover a melhoria da qualidade, bem como a formação dos estudantes. Avalia, entre outros aspectos, ensino, pesquisa, extensão, responsabilidade social, desempenho dos alunos, gestão da instituição, corpo docente e instalações (INEP, 2009).

5.2 Perspectivas e indicadores

O SINAES disponibiliza padrões definidos como dimensões que podem, no SISCORE, ser usadas de forma análoga às perspectivas, ou seja, a “Dimensão Docentes” é usada como “Perspectiva Docentes”. Os indicadores que o MEC sugere têm as medidas

estabelecidas através de estudos de como eles serão mensurados.

Através de inspeção no SINAES foi possível identificar as dimensões que deram origem às perspectivas aplicadas ao uso do SISCORE. Por exemplo, alguns indicadores da dimensão Docentes incluem (i) Produção acadêmica/docentes, (ii) aluno tempo integral/professor, (iii) grau de envolvimento como pós-graduação, (iv) grau de envolvimento com pesquisa e (v) grau de envolvimento com extensão. As perspectivas e indicadores identificados no padrão SINAES estão relacionados no Quadro 10.

Perspectiva	Indicador
Pós-Graduação	Atividades científicas
	Conceitos da CAPES
	Atuação profissional dos egressos
	Publicações
Docentes	Conceitos da CAPES no Pós-graduação stricto sensu
	Índice de Qualificação do Corpo Docente.
	Produção acadêmica/docentes
	Aluno tempo integral/professor
	Grau de envolvimento como pós-graduação, pesquisa e extensão
Técnico-Administrativa	Aluno tempo integral/funcionário técnico-administrativo
Infra-estrutura física	Nível de satisfação dos usuários sobre as equipamentos informáticos
	Nível de satisfação dos usuários sobre as instalações, biblioteca e laboratórios
Planejamento e avaliação	Projeto Pedagógico
	Relatórios de auto-avaliação
	Ações decorrentes das conclusões da auto-avaliação
	Nº de eventos e seminários de difusão dos processos de auto-avaliação
Egressos	TSG - Taxa de Sucesso na Graduação
	GPE - Grau de Participação Estudantil
	Tempo médio de conclusão do curso
	Aluno tempo integral/professor
	Aluno tempo integral/funcionário técnico-administrativo
Sustentabilidade financeira	Relação orçamento/gastos
	Relação ingressantes/concluintes
	Relação docentes em capacitação/docentes
	Relação dos técnicos-administrativos em capacitação/ capacitados

Quadro 10 – Tabela de perspectivas e indicadores

Fonte: Adaptado de SINAES (2009)

Para o indicador “produção acadêmica/docentes”, presente no Quadro 10, a medida para mensurá-la é a média do número de publicações por estudantes e professores. Já o indicador “Aluno tempo integral/professor” é medido através do percentual de presença que os estudantes têm por professor em sala de aula. De outra maneira o indicador “Grau de envolvimento como pós-graduação” se apresenta de duas formas: (i) quantos estudantes formados na IES inscrevem-se nos cursos de pós-graduação e (ii) quantos formados em outras instituições mostram interesse pelos cursos.

O uso dos indicadores fornecidos pelo MEC, mesmo que com valores fictício, serviu para verificar o quanto o SISCORE pode gerenciar esses dados pré-definidos e como a ferramenta se comporta com a atualização, acompanhamento e demonstração de resultados.

5.3 Tela de acesso

Toda aplicação, seja ela *desktop* ou Web, por motivos de controle e segurança, deve proporcionar acesso através da identificação do usuário que deseja acessá-la. A Figura 28 apresenta a tela de acesso para todas as funcionalidades disponíveis na aplicação.

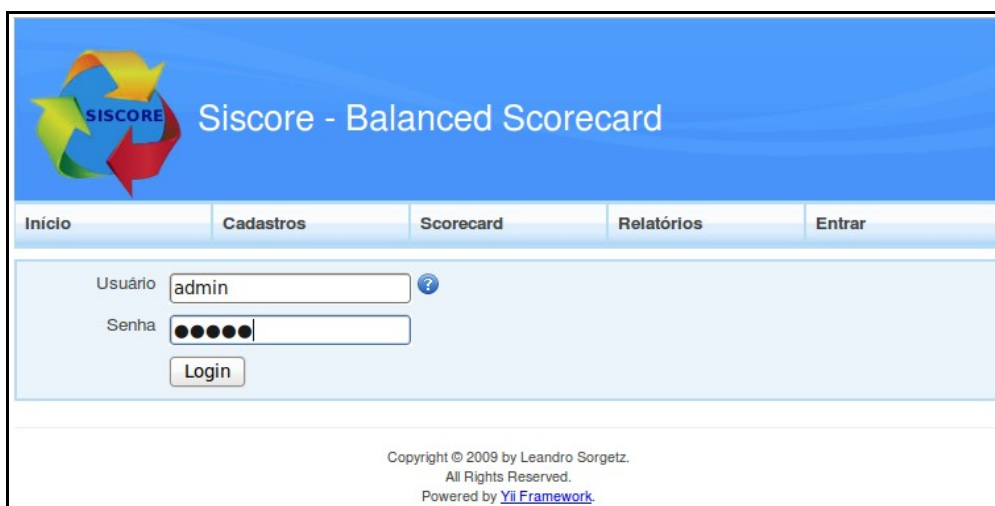


Figura 28 – Tela de Login

Fonte: o autor

O usuário que deseja acessar o SISCORE deve estar cadastrado com as permissões relativas às ações que deseja realizar. Pode-se observar que, de acordo com os requisitos de

Niven (2003), a tela dispõe de um botão para obter informações úteis para oferecer ajuda ao usuário. Esse botão quando acionado apresenta um *popup*³⁷ implementado através da extensão Highslide JS³⁸ como ilustrado na Figura 29.

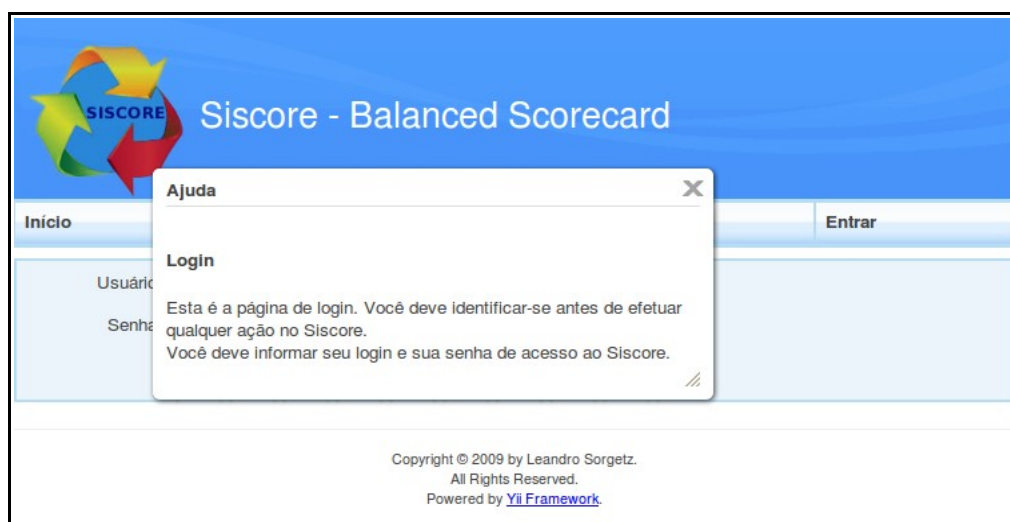


Figura 29 – Popup de ajuda

Fonte: o autor

5.4 Tela inicial

Após a identificação o usuário dispõe de uma tela inicial ilustrada na Figura 30 onde encontram-se três figuras que possibilitam ao usuário remeter-se, diretamente ao Gráfico do Indicador Padrão, ao Mapa Estratégico e à Alimentação do Indicador ao qual ele é responsável.

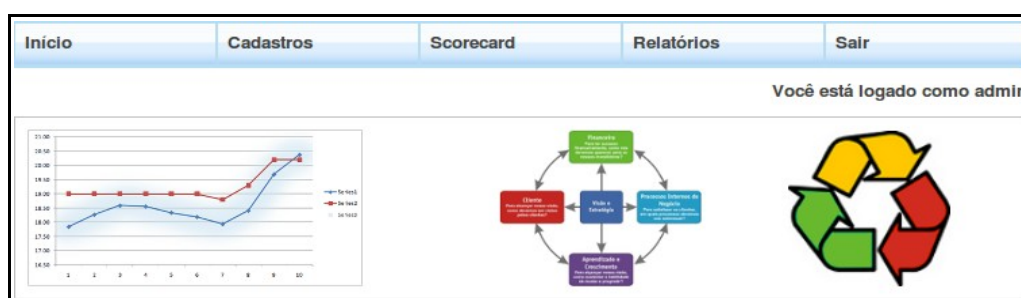


Figura 30 – Tela Inicial

Fonte: o autor

³⁷ Janela que abre dentro do navegador.

³⁸ Visualizador *opensource* escrito em javascript para a visualização de mídias e galerias (<http://highslide.com>).

Nesta figura ainda observa-se que o menu de opções que está dividido em Início, Cadastros, Scorecard, Relatórios e Sair. Todas essas funcionalidades são restritas às permissões disponíveis para cada usuário.

5.5 Gerenciamento dos cadastros

Um sistema como o SISCORE depende de uma série de cadastros para registro de todos os dados necessários à implementação de um BSC. Esses cadastros armazenam todas as informações pertinentes à execução de um plano estratégico gerencial. Quase todas as tabelas são cadastros no banco de dados do SISCORE. A Figura 31 ilustra o módulo *admin* que disponibiliza a gestão dos registros do cadastro de indicadores e apresenta todas as outras quatro opções básicas.

Gestão de indicadores						
Ir à página: < Anterior 1 2 Próximo >						
Nome	status	Unidade de Medida	Plano de ação	Responsável	Indicador Pai	Ações
Aluno tempo integral/funcionário técnico-adm		Unidade	Qualificar o nível técnico-administrativo	Antônio Carlos da Silveira	Aluno tempo integral/funcionário técnico-adm	   
Grau de satisfação usuários da biblioteca		Unidade	Melhorar a estrutura física	Antônio Carlos da Silveira	Grau de satisfação usuários da biblioteca	   
Atuação profissional dos egressos		Unidade	Promover cursos de pós-graduação	Antônio Carlos da Silveira	Atuação profissional dos egressos	   
Qtde Projetos pedagógicos Institucionais		Unidade	Aumentar a Sustentabilidade financeira	Antônio Carlos da Silveira	Qtde Projetos pedagógicos Institucionais	   
Ir à página: < Anterior 1 2 Próximo >						

Figura 31 – Modelo de tela de gestão – indicadores

Fonte: o autor

Os principais elementos que compõe a maioria das gestões dos cadastros disponíveis no SISCORE estão dispostos na Figura 31. A gestão de um cadastro é composto pelo nome da gestão em questão, três botões visíveis no canto superior direito que remete o usuário à listagem dos indicadores (*list*), inclusão de indicadores (*create*) e ajuda (*help*), respectivamente. Na linha seguinte está a barra de navegabilidade entre as páginas, disponível também ao final de cada página.

Observa-se que além dos três botões de operações aparecem mais dois que são o de apresentação do gráfico do indicador em questão e o de alimentação do mesmo. Essas funcionalidades serão melhor apresentadas na próxima sessão.

Após a apresentação dos dados mais relevantes encontra-se a coluna de ações possíveis para cada registro da tabela em questão. São visíveis através de três botões que permitem o usuário alterar (*update*) um registro, consultar seus dados (*show*) ou excluí-lo (*delete*). Uma vez que o usuário clica no botão editar, será apresentada uma tela semelhante ao exemplo da Figura 32.

Edita pessoa Antônio Carlos da Silveira

* Campo obrigatório.

Nome * Antônio Carlos da Silveira

Equipe * Coordenações

Cargo * Gerente

Identidade 9039465159

Tipopessoa * Física

Telefone 2147483647

E Mail acarlos@faccat.br

Site fit.faccat.br/~acarlos

Salvar

Figura 32 – Modelo de tela Edição – pessoa

Fonte: o autor

Através da tela de edição da Figura 32 o usuário também pode acessar a listagem, inclusão, gestão e ajuda. A tela de inclusão é muito semelhante tendo como diferença apenas nos botões de navegabilidade. Ao clicar sobre o botão listagem o usuário obtém uma lista como a que é mostrada na Figura 33.

Lista de equipes

Nome: Coordenações
Observação: Equipe responsável pelas informações da coordenação

Nome: Secretaria
Observação: Equipe responsável pelas informações da secretaria

Nome: Legislação
Observação: Equipe responsável pelo setor de legislação

Nome: Professores
Observação: Equipe responsável pelas informações dos professores

Figura 33 – Modelo de tela de listagem – equipes

Fonte: o autor

Assim como as demais telas a da Figura 33 também dispõe de alguns botões de navegabilidade. Outra funcionalidade, ilustrada na Figura 34, é acionada ao clicar no botão mostrar da Gestão de Perspectivas que, por sua vez mostra os dados do registro escolhido.

Mostra perspectiva Pós-Graduação	
Nome	Pós-Graduação
Estratégia	Aumentar a qualidade dos cursos de Pós
Descrição	
Cor	

Figura 34 – Modelo de tela de listagem – equipes

Fonte: o autor

Em todos os momentos, na gestão de um determinado cadastro, o usuário dispõe de recursos que o remetem diretamente para os demais funcionalidades disponíveis para cada cadastro. Isso atende a um dois requisitos de Niven (2003) que é o de facilitar a navegabilidade e tornar a ferramenta mais produtiva.

5.6 Criação de um BSC com o SISCORE

A implementação de um BSC através do SISCORE requer uma sequencia de procedimentos que deve ser seguida. Inicia com a preparação das tabelas e cadastros necessários para operações específicas a um BSC.

Inicialmente todos os dados deve ser incluídos nas tabelas periféricas que são necessárias para a inclusão dos dados dos cadastros que as utilizam. Essas tabelas são Tipos de Logradouro, Estados, Cidades, Cargos, Unidades de Medida, Tipo de Pessoa e Permissão, Estratégia, Periodicidade, Unidade de Medida e Sinalizador.

A partir desse momento os cadastros que dependem das tabelas acima relacionadas podem ser criados. São eles, o cadastro de Cidade, Grupo, Perspectiva, Plano de Ação, Pessoa, Indicador e Iniciativa, essa ordem deve ser seguida por causa da dependência que existe entre eles.

Para que se possa acompanhar a evolução de um indicador é necessário alimentá-lo, o que que significa inserir os dados relativos ao seu desempenho. Esse desempenho somente

poderá ser medido mediante a inclusão das metas estabelecidas para esse indicador. A Figura 35 ilustra a tela de inclusão de metas para um determinado indicador.

Campos com * são obrigatórios.

Indicador *

Data *

Valor *

Figura 35 – Metas por indicador

Fonte: o autor

Para o indicador “Grau de satisfação usuários da biblioteca”, visto na Figura 37, foi determinado que ele dever atingir a marca de 80% até a data de 27 de novembro de 2009. Feito isso, o indicador está preparado para receber movimentações através da funcionalidade “Alimentar indicador” ilustrado na Figura 36.

Estratégia

Perspectiva

Plano De Ação

Indicador *

Data *

Histórico

Observação

Valor

Figura 36 – Alimentar indicador

Fonte: o autor

A localização de um indicador específico pode tornar-se uma tarefa difícil considerando que ele pode estar no meio de uma lista muito extensa. Por isso, a alimentação de indicadores, ilustrada na Figura 36, possui filtros para facilitar essa seleção. Inicialmente o usuário informa a estratégia a que o indicador desejado pertence. Após isso a

aplicação lista as perspectivas relacionadas a essa estratégia que. A seleção de uma perspectiva filtra os planos de ação. Por fim, após selecionado o plano de ação, uma lista reduzida de indicadores é disponibilizada para seleção. Este é o estágio final da inclusão de dados. A partir desse momento, já é possível verificar os resultados obtidos.

5.7 Monitoramento dos resultados

A tarefa de verificar os resultados está diretamente relacionada ao analista de negócio. Este, por sua vez, monitora o BSC através de gráficos que são ilustrados e detalhados na próxima Subseção. Além dos gráficos, é possível obter um panorama geral de como está o desempenho dos indicadores através do gestão de indicadores ilustrada na Figura 37.

Nome	status	Unidade de Medida	Plano de ação	Responsável	Indicador Pai	Ações
Aluno tempo integral/funcionário técnico-adm		Unidade	Qualificar o nível técnico-administrativo	Antônio Carlos da Silveira	Aluno tempo integral/funcionário técnico-adm	    
Grau de satisfação usuários da biblioteca		Unidade	Melhorar a estrutura física	Antônio Carlos da Silveira	Grau de satisfação usuários da biblioteca	    
Grau de satisfação usuários dos laboratórios		Unidade	Melhorar a estrutura física	Regina Soares Peroni	Grau de satisfação usuários dos laboratórios	    
Grau de satisfação usuários de equip. inform.		Unidade	Melhorar a estrutura física	Regina Soares Peroni	Grau de satisfação usuários de equip. inform.	    
Produção acadêmica/docentes		Unidade	Melhorar o nível docente	Antônio Carlos da Silveira	Produção acadêmica/docentes	    
Grau de envolvimento com pesquisa		Unidade	Melhorar o nível docente	Antônio Carlos da Silveira	Grau de envolvimento com pesquisa	    
Grau de envolvimento como pós-graduação		Unidade	Melhorar o nível docente	Antônio Carlos da Silveira	Grau de envolvimento como pós-graduação	    
Publicações		Unidade	Promover cursos de pós-graduação	Regina Soares Peroni	Publicações	    
Atuação profissional dos egressos		Unidade	Promover cursos de pós-graduação	Antônio Carlos da Silveira	Atuação profissional dos egressos	    
Qtde Projetos pedagógicos Institucionais		Unidade	Aumentar a Sustentabilidade financeira	Antônio Carlos da Silveira	Qtde Projetos pedagógicos Institucionais	    

Figura 37 – Gestão de indicadores

Fonte: o autor

A situação de cada indicador é apresentada pela coluna “*status*” vista na Figura 37. A cor amarela significa que o indicador está próximo do limite pré-estabelecido, a cor verde que ele se encontra acima da expectativa, a azul indica estar dentro do esperado e vermelho que está aquém da expectativa. Através dessa gerência o usuário pode visualizar o gráfico de desempenho de um determinado indicador. Essa ação é representada pelo penúltimo ícone da

coluna “Ações”. Outra forma de verificar o status dos indicadores é através do mapa estratégico disposto na Figura 38.

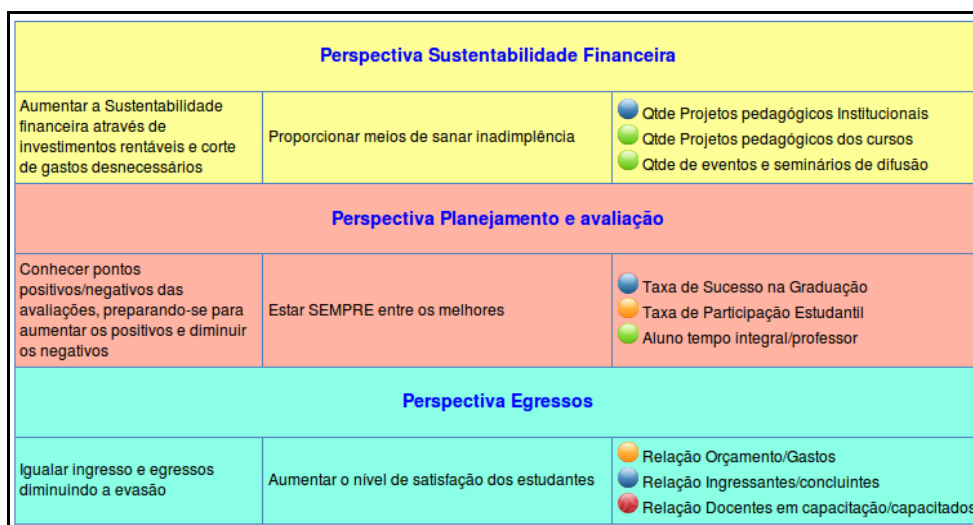


Figura 38 – Mapa Estratégico

Fonte: o autor

Assim como na gestão de indicadores ilustrada na Figura 37, é possível verificar a situação de cada indicador como visto na Figura 38. Este mapa distribui os indicadores explicitando as suas associações de acordo com um determinado plano de ação, estratégia e perspectiva. Por exemplo, na Figura 38, a perspectiva “Egressos” contém o indicador “Relação Orçamento/Gastos” que pertence à estratégia “Aumentar o nível de satisfação dos estudantes” que, por sua vez está associada ao plano de ação “Igualar ingressos e egressos diminuindo a evasão”.

5.8 Visualizando resultados através de gráficos

A visualização das informações por meio de gráficos, facilita a análise, porque agrupa os elementos e proporciona uma macrovisão do negócio. O Gráfico 4 apresenta a evolução do indicador “Atividades científicas” em relação às metas estabelecidas para ele.

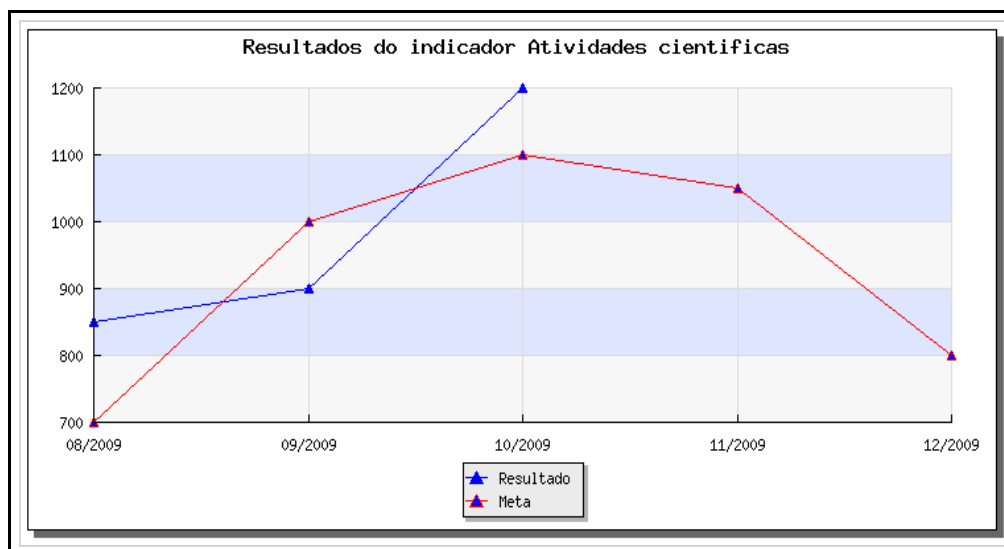


Gráfico 4 – Desempenho de um indicador

Fonte: o autor

A linha azul, no gráfico do Gráfico 4, mostra os resultados obtidos no indicador, que vão de agosto a outubro de 2009, período que compreende o início do ciclo deste indicador até a data de seu último lançamento. Já a linha vermelha é referente a meta que foi estabelecida para esse indicador, que vai de agosto a dezembro de 2009.

Pode-se observar que o indicador, que está além da meta estabelecida para o primeiro mês, apresenta uma queda no segundo e tem novamente um aumento no terceiro mês. Isso denota que o indicador está dentro da média. No Gráfico 5 está disposta a evolução de três indicadores escolhidos pelo usuário.

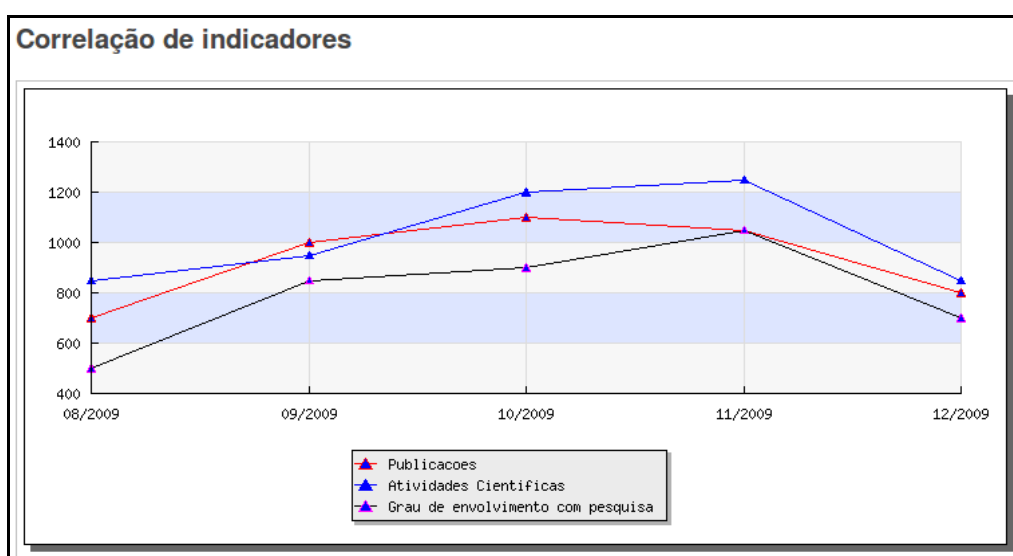


Gráfico 5 – Correlação de diferentes indicadores

Fonte: o autor

Com resultados semelhantes aos apresentados no Gráfico 5 é possível identificar possíveis relações entre os indicadores. Essa relação de causa e efeito, inicialmente apenas imaginada, somente é confirmada por meio de constatações baseadas nos resultados atingidos por cada indicador. Assim, de uma maneira simples e direta, o Analista de Negócio acompanha a evolução de todos os indicadores cadastrados.

6 CONCLUSÕES

Este trabalho apresentou o SISCORE, uma ferramenta Web para a implementação do *Balanced Scorecard*, desenvolvido utilizando padrões de projeto combinados com ferramentas livres para o desenvolvimento de aplicações Web.

A pesquisa por ferramentas que propõem implementar o BSC e a análise dos exemplos e casos apresentados na proposta de Kaplan e Norton (1997), mostram o quanto o foco dessa metodologia está voltado a grandes empresas. Entretanto essa ferramenta de planejamento estratégico é perfeitamente aplicável a pequenas e médias empresas.

A escolha por um ambiente cujas tecnologias utilizadas são livres e de código aberto foi de grande importância para possibilitar um aplicativo de custo mais acessível do que as propostas disponíveis no mercado de *software* proprietário.

Dentre as opções de ferramentas existente para desenvolvimento, destaca-se o uso do *framework* Yii. Essa escolha tornou-se essencial para facilitar a implementação e auxiliar na aplicação do padrão de projeto MVC. A geração automática das classes pelo Yii permitiu focar o esforço do desenvolvimento nas regras de negócio. Além disso, pode-se contar com uma série de itens de segurança e recursos de visualização avançados. As extensões que a comunidade desenvolve e disponibiliza no *site* do Yii também são muito úteis, pois contemplam necessidades comuns a maioria das aplicações Web.

O SISCORE foi desenvolvido consultando profissionais da área acadêmica das Faculdades Integradas de Taquara. Esses profissionais aplicam o BSC através do uso de outras ferramentas, tais como planilhas eletrônicas, editores de texto, entre outros. O contato com essas pessoas despertou seu interesse para suprir uma demanda pelo uso de uma ferramenta de BSC no auxílio do aprendizado das disciplinas dos cursos de Administração, mais especificamente em disciplinas de cunho estratégico. Profissionais da área de TI desta IES também mostraram interesse em implementar o BSC com o auxílio do SISCORE, motivados pelo caso de sucesso da Grandene. Este caso é apresentado na Seção 3 deste trabalho.

Futuramente ainda serão incorporadas diversas funcionalidades não previstas inicialmente. Dentre elas incluem-se (i) acompanhamento, visualização de gráficos e alimentação dos indicadores através de dispositivos móveis, tais como *smartphones* e PDAs *Personal Digital Assistant* (Assistentes Digitais Pessoais); (ii) opções de monitoramento por

meio diferentes tipos de gráficos; (iii) identificação de relações de causa e efeito entre indicadores de forma automática; (iv) integração com demais sistemas de gestão, sejam administrativos ou estratégicos.

Outro passo importante será preparar a documentação do SISCORE e disponibilizá-la através do site SOURCEFORGE.NET. Essa iniciativa conta com a adesão da comunidade desenvolvedora de *Software* Livre e visa o crescimento, a disseminação e a popularização dessa proposta.

Por fim conclui-se que o emprego de todas essas metodologias, técnicas, ferramentas e padrões permitiram atingir o objetivo desse trabalho, resultando numa ferramenta útil, prática e de fácil utilização e que permite implementar um BSC de acordo com a metodologia proposta por Kaplan e Norton (1997).

7 REFERÊNCIAS

ADEMPIERE. Disponível em: <<http://www.adempiere.org/>>. Acesso em: 23 mai. 2009.

BERINGER Consulting. Disponível em: <<http://www.beringerconsulting.net/>>. Acesso em: 23 mai. 2009.

BLUEOXYGEN. Disponível em: <<http://www.blueoxygen.org/>>. Acesso em: 23 mai. 2009.

BSPG – *An open source balanced scorecard*. Disponível em: <<http://bspg.sourceforge.net/>>. Acesso em: 23 mai. 2009.

BUDD, Andy; MOOL, Cameron; COLLISON, Simon. **Criando páginas Web com CSS**. São Paulo: Pearson Prentice Hall, 2006.

CAKEPHP. *The PHP rapid development framework*. Disponível em: <<http://www.cakephp.net/>>. Acesso em: 27 ago. 2007.

CAKEPHP Brasil. Disponível em: <<http://blog.cakephp-brasil.org/>>. Acesso em: 23 mai. 2009.

COSTA, Ana Paula Paulino da. **Balanced scorecard: conceitos e guia de implementação**. São Paulo: Atlas, 2006.

GAMMA, Erich; HELM, Richard; JOHNSON, Ralf; VLISSIDES, John. Trad. Salgado, Luiz A. Meirelles. **Padrões de projeto – soluções reutilizáveis de software orientado a objeto**. Porto Alegre: Bookmann, 2000.

GRENDENE. Disponível em <<http://www.grendene.com.br>>. Acesso em: 23 mai. 2009.

GUTMANS, Andi; BAKKEN, Stig Sæther; Rethans, Derick. **PHP 5 – Programação poderosa**. Rio de Janeiro: Alta Books, 2005.

FOWLER, Martin; SCOTT, Kendall. **UML essencial – um breve guia para a linguagem-padrão de modelagem de objetos**. Porto Alegre: Bookman, 2000.

FOWLER, Martin. **Padrões de arquitetura de aplicações corporativas**. Porto Alegre: Bookman, 2006.

HIKAGE, Oswaldo Keiji; SPINOLA, Mauro de Mesquita; LAURINDO, Fernando José Barbin. *Software de balanced scorecard: proposta de um roteiro de implantação*. **Produção**, São Paulo, v. 16, n. 1, p. 140-160, Jan./Abr. 2006.

GOOD, A. Nathan. *The future of PHP*. Disponível em: <<http://www.ibm.com/developerworks/opensource/library/os-php-future/>>. Acesso em: 23 mai. 2009.

IBM. *Cognos Business Intelligence and Financial Performance Management*. Disponível em <<http://www-01.ibm.com/software/data/cognos/>>. Acesso em: 30 ago. 2009.

INEP. **Instituto Nacional de Estudos e Pesquisas Educacionais – Sistema Nacional de Avaliação da Educação Superior**. Disponível em <<http://www.inep.gov.br/superior/SINAES/>>. Acesso em: 30 ago. 2009.

INTERACT Solutions. *Manual Strategic Adviser – Explorer*. Disponível em <<http://www.interact.com.br>>. Acesso em: 28 nov. 2007.

KAPLAN, Robert S.; NORTON, David P. *A estratégia em ação: balanced scorecard*. – 16. ed. Rio de Janeiro: Campus, 1997.

MORAIS, Roberto Tadeu Ramos. *Planejamento estratégico: um bem ou mal necessário?* Taquara: Faccat, 2004.

MÜLLER JUNIOR, Werner Arthur. *Balanced Scorecard – gestão de informações gerenciais para empresa calçadista*. Taquara: FACCAT, 2001.

MINETTO, Elton Luís. *Frameworks para desenvolvimento em PHP*. São Paulo: Novatec, 2007.

NIVEN, P.R. *Balanced scorecard step-by-step for government and nonprofit agencies*. New Jersey: John Wiley & Sons, 2003.

OPENBRAVO. *Web-baser open source ERP solution*. Disponível em: <<http://www.openbravo.com>>. Acesso em: 30 jun. 2009.

PALLADIUM. *Executing Strategy*. Disponível em: <<http://www.thepalladiumgroup.com>>. Acesso em: 23 mai. 2009.

PENTAHO. *Open source business intelligence*. Disponível em: <<http://pentaho.com/>>.

Acesso em: 23 mai. 2009.

PHPFRAMEWORKS. Disponível em: <<http://www.phpframeworks.com/>>. Acesso em: 23 mai. 2009.

PHP. *Hypertext preprocessor*. Disponível em: <<http://php.net/>>. Acesso em: 23 mai. 2009.

PRESSMAN, Roger S. **Engenharia de Software**. 5.ed. Rio de Janeiro: McGraw-Hill, 2002.

RAGGETT, Dave. **Getting started with HTML**. Disponível em <<http://www.w3.org/MarkUp/Guide/>>. Acesso em: 12 nov. 2007.

SCOTT, Kendall. **O Processo unificado exemplificado**. Porto Alegre: Bookman, 2003.

SILBERSCHARTZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. **Sistema de banco de dados**. 3.ed. São Paulo: Makron Books, 1999.

SOARES, Wallace. **Ajax – Assynchronous javascript and xml – guia prático**. São Paulo: Érica, 2006.

SOURCEFORGE.NET – *Find and build open source software*. Disponível em <<http://sourceforge.net/>>. Acesso em: 23 mai. 2009.

SUCESU-RS – **Sociedade dos usuários de informática e telecomunicações do Rio Grande do Sul**. Disponível em <<http://www.rs.sucesu.org.br/>>. Acesso em: 23 mai. 2009.

W3C – *World Wide Web Consorcium*. Disponível em <<http://www.w3.org>>. Acesso em: 18 nov. 2007.

YII– *Web Programming Framework*. Disponível em <<http://www.yiiframework.com/>>. Acesso em: 14.jul. 2009.

APÊNDICES

APÊNDICE A – Detalhamento de um caso de uso

Detalhamento do Caso de Uso: UC001 – Alimentar indicadores

1 Descrição

Este caso de uso descreve quais os passos necessários para um usuário alimentar um determinado indicador.

2 Ator

Responsável

3 Pré-Condições

O usuário deve estar logado no sistema com o perfil Responsável para alimentar um determinado indicador.

4 Fluxo de eventos

4.1 Fluxo Básico

Passos:

[P01] Caso de Uso inicia quando o usuário seleciona a opção “Alimentar Indicador” [A01]

[P02] Sistema apresenta interface [SI01] [RN01]

[P03] Usuário confirma ação [A02] [A03]

[P04] Sistema retorna ao passo [P01]

4.2 Fluxo Alternativo

[A01] – Usuário inválido

Usuário não se identificou com permissões adequadas

Passos:

[P01.A01] – Sistema envia uma mensagem informando que usuário não tem permissão para acessar esse Caso de Uso

[P02.A01] – Sistema encerra o Caso de Uso

[A02] – Dados inválidos

Pré-condições: Usuário formou algum dado fora do seu domínio e/ou obrigatoriedade

Passos:

[P01.A02] – Sistema volta ao [P02] indicando campos inválidos e solicitando que usuário os corrija

[A03] – Operador de Caixa cancela ação de Abrir Caixa

Pré-condições: Operador de Caixa cancelou a ação

Passos:

[P01.A03] – Sistema encerra Caso de Uso

6 Regras de Negócio

[RN01] – Detalhamento dos campos exibidos na Interface [SI01] do passo [P02]

Campo	Domínio	Obrigatório	Observações
Estratégia	Lista	Sim	
Perspectiva	Lista	Sim	Relaciona somente perspectivas associadas à estratégia selecionada
Plano de ação	Lista	Sim	Relaciona somente planos de ação associados à perspectiva selecionada
Indicador	Lista	Sim	Relaciona indicadores associados ao plano de ação selecionado
Data	Data	Sim	
Histórico	Texto	Não	
Observação	Texto	Não	
Valor	Numérico	Sim	Positivo

7 Interfaces de *software*

[SI01] – Alimentar indicador

Campos com * são obrigatórios.

Estratégia *	Escolha uma Estratégia
Perspectiva *	Escolha uma Perspectiva
Plano De Ação *	Escolha um Plano de ação
Indicador *	Escolha um indicador
Data *	
Histórico	
Observação	
Valor *	