

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
II CURSO DE ESPECIALIZAÇÃO EM BANCO DE DADOS**

**IGOR PEDROSO GUIMARÃES**

**SOFTWARE PARA MONITORAMENTO DE OPERAÇÕES EM  
BANCOS DE DADOS**

**MONOGRAFIA DE ESPECIALIZAÇÃO**

**PATO BRANCO  
2017**

**IGOR PEDROSO GUIMARÃES**

**SOFTWARE PARA MONITORAMENTO DE OPERAÇÕES EM  
BANCOS DE DADOS**

Trabalho de Conclusão de Curso, apresentado ao II Curso de Especialização em Banco de Dados, da Universidade Tecnológica Federal do Paraná, campus Pato Branco, como requisito parcial para obtenção do título de Especialista.

Orientadora: Profa. Beatriz Terezinha Borsoi

**PATO BRANCO  
2017**



---

## TERMO DE APROVAÇÃO

*SOFTWARE* PARA MONITORAMENTO DE OPERAÇÕES EM BANCO DE DADOS.

por

IGOR PEDROSO GUIMARÃES

Este Trabalho de Conclusão de Curso foi apresentado em 06 de abril de 2017 como requisito parcial para a obtenção do título de Especialista em Banco de Dados. O(a) candidato(a) foi arguido(a) pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

---

Beatriz Terezinha Borsoi  
Prof.(a) Orientador(a)

---

Geri Natalino Dutra  
Membro titular

---

Vinicius Pegorini  
Membro titular

“O Termo de Aprovação assinado encontra-se na Coordenação do Curso”

## RESUMO

GUIMARÃES, Igor Pedroso. Software para monitoramento de operações em bancos de dados de clientes. 2017. 48 f. Monografia (II Curso de Especialização em Banco de Dados) – Universidade Tecnológica Federal do Paraná. Pato Branco, 2017.

A tomada de decisão no ambiente de negócio é cada vez mais dependente de dados gerados pelas operações envolvidas no negócio e obtidas de fontes externas. Os dados gerados internamente têm se tornado cada vez mais relevantes para apoio à tomada de decisão pelo volume e diversidade de dados gerados e surgem conceitos, técnicas e ferramentas para auxiliar na análise desses dados. O termo Business Intelligence (BI), inteligência de negócios, representa o processo de coleta, organização, análise, distribuição e monitoramento de dados que geram informação para suporte à gestão de negócios. A inteligência de negócios pode ser utilizada nos diversos níveis de decisão organização e nos mais variados setores das empresas. Os DataBase Administrator (DBA), administradores de bancos de dados, podem utilizar-se de dados obtidos do monitoramento de operações realizadas em bases de dados de clientes como suporte à tomada de decisão em um processo de BI, visando otimizar o tempo e os recursos das operações realizadas, evitar e prevenir erros. Considerando esse contexto, foi desenvolvido um sistema para monitoramento em tempo real de operações realizadas em base de dados de clientes de um sistema de gestão, visando reduzir o tempo de tomada decisão de DBAs. O sistema desenvolvido, que é web, apresenta indicadores sobre a consistência dos serviços hospedados em um conjunto de servidores de forma personalizada.

**Palavras-chave:** DataBase Administrator. Monitoramento de operações de Banco de Dados. Hospedagem de serviços.

## **ABSTRACT**

GUIMARÃES, Igor Pedroso. Software for monitoring operations in clientes databases. 2017. 48 p. Monography (II Specialization Course in Database) - Federal University of Technology - Parana. Pato Branco, 2017.

Decision making in business environments is increasingly dependent on data generated by operations involved in the business and obtained from external sources. Data generated internally have become increasingly relevant to support decision making by the volume and diversity of data generated and concepts, techniques and tools to assist in the analysis of these data. The term Business Intelligence (BI), represents the process of collecting, organizing, analyzing, distributing and monitoring data that generates information to support business management. Business intelligence can be used at the various organizational decision-making levels and across a wide range of business sectors. DataBase Administrator (DBA), database administrators, can use data obtained from the monitoring of operations performed in customer databases as support for decision making in a BI process, in order to optimize the time and resources Avoiding and preventing errors. Considering this context, a system was developed for real-time monitoring of operations carried out in the database of clients of a management system, aiming to reduce the decision-making time of DBAs. The developed system, which is web-based, presents indicators on the consistency of services hosted in a set of servers in a personalized way.

Keywords: Database Administrator. Database monitoring operations. Service hosting.

## LISTA DE FIGURAS

<b>Figura 1 – Habilitar o IIS .....</b>	<b>28</b>
<b>Figura 2 – Visual Studio 2015.....</b>	<b>28</b>
<b>Figura 3 – Mockup: dashboard da aplicação.....</b>	<b>31</b>
<b>Figura 4 – Mockup: detalhes do serviço.....</b>	<b>31</b>
<b>Figura 5 – Diagrama: processo do sistema.....</b>	<b>32</b>
<b>Figura 6 – Diagrama: processo do sistema detalhes monitor .....</b>	<b>33</b>
<b>Figura 7 – Diagrama de Entidade e Relacionamento .....</b>	<b>34</b>
<b>Figura 8 – Tela de login do sistema .....</b>	<b>36</b>
<b>Figura 9 – Tela principal do sistema .....</b>	<b>37</b>
<b>Figura 10 – Rotinas do SPService.....</b>	<b>37</b>
<b>Figura 11 – Tela de detalhes do serviço.....</b>	<b>38</b>
<b>Figura 12 – Tela de validação do login (usuário e senha).....</b>	<b>41</b>

## LISTA DE QUADROS

<b>Quadro 1 – Ferramentas e tecnologias utilizadas .....</b>	<b>17</b>
--	-----------

## LISTAGENS DE CÓDIGO

Listagem 1 – HTML página de login.....	39
Listagem 2 – Código da função de login.....	40
Listagem 3 – Método visualização detalhada dos serviços .....	42
Listagem 4 – CSS do Bootstrap .....	42
Listagem 5 – Funções coletoras de índice.....	44



## LISTA DE SIGLAS

BI	<i>Business Intelligence</i>
BPMN	<i>Business Process Management Notation</i>
CASE	<i>Computer Aided Software Engineering</i>
CSS	<i>Cascading Style Sheets</i>
DBA	<i>DataBase Administrator</i>
DER	<i>Diagramas de Entidade Relacionamento</i>
FTP	<i>File Transfer Protocol</i>
GNU	<i>General Public License</i>
HTML	<i>HyperText Markup Language</i>
IDE	<i>Integrated Development Environment</i>
IIS	<i>Internet Information Services</i>
MIT	<i>Massachusetts Institute of Technology</i>
MVC	<i>Model-View-Controller</i>
PHP	<i>PHP Hypertext Preprocessor</i>
RAD	<i>Rapid Application Development</i>
RGM	<i>Relatório Gerencial Mensal</i>
SGBD	<i>Sistema Gerenciado de Banco de Dados</i>
SSMS	<i>SQL Server Management Studio</i>
VB	<i>Visual Basic</i>
WCF	<i>Windows Communication Foundation</i>
XML	<i>eXtensible Markup Language</i>

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>10</b>
1.1 CONSIDERAÇÕES INICIAIS .....	10
1.2 OBJETIVOS .....	12
<b>1.2.1 Objetivo Geral</b> .....	<b>12</b>
<b>1.2.2 Objetivos específicos</b> .....	<b>12</b>
1.3 JUSTIFICATIVA.....	12
1.4 ESTRUTURA DO TRABALHO .....	13
<b>2 REFERENCIAL TEÓRICO</b> .....	<b>14</b>
<b>3 MATERIAIS E MÉTODOS</b> .....	<b>17</b>
3.1 MATERIAIS.....	17
<b>3.1.1 Visual Studio 2015</b> .....	<b>13</b>
<b>3.1.2 Microsoft Visual Basic.Net</b> .....	<b>14</b>
<b>3.1.3 Asp Net</b> .....	<b>15</b>
<b>3.1.4 .Net Framework</b> .....	<b>15</b>
<b>3.1.5 Case Studio</b> .....	<b>15</b>
<b>3.1.6 HTML 5</b> .....	<b>16</b>
<b>3.1.7 BootsStrap</b> .....	<b>18</b>
<b>3.1.8 SQL Server 2014 Express</b> .....	<b>19</b>
<b>3.1.9 Microsoft Sql Server Management</b> .....	<b>20</b>
<b>3.1.10 Ajax</b> .....	<b>20</b>
<b>3.1.11 JQuery</b> .....	<b>21</b>
<b>3.1.12 JavaScript</b> .....	<b>21</b>
<b>3.1.13 Css3</b> .....	<b>22</b>
<b>3.1.14 Bizagi</b> .....	<b>22</b>
<b>3.1.15 IIS</b> .....	<b>22</b>
<b>3.2 MÉTODOS</b> .....	<b>25</b>
<b>4 RESULTADO</b> .....	<b>30</b>
4.1 DESCRIÇÃO DO SISTEMA.....	30
4.2 MODELAGEM DO SISTEMA.....	30
4.3 DESCRIÇÃO DO SISTEMA.....	30
4.4 IMPLEMENTAÇÃO DO SISTEMA .....	38
<b>CONCLUSÃO</b> .....	<b>43</b>
<b>REFERÊNCIAS</b> .....	<b>45</b>

## 1 INTRODUÇÃO

Neste capítulo serão apresentadas as considerações iniciais com a inserção do trabalho proposto no contexto conceitual e os objetivos e a justificativa do trabalho, bem como a estrutura dos capítulos subsequentes.

### 1.1 CONSIDERAÇÕES INICIAIS

Um serviço do Sistema Operacional Windows é uma aplicação que inicia quando o Windows é carregado, ou sob demanda, e executa em segundo plano, ou seja, em *background*, enquanto o Windows estiver em execução (PAULINO, 2009). Essas aplicações são responsáveis pela realização de tarefas essenciais, como serviços de impressão e de *e-mail*. Ideia semelhante à forma de realização desses serviços pode ser utilizada para rotinas em bancos de dados que são realizadas sem a interferência do usuário e que podem ser executadas em período de inatividade do sistema, como, durante a madrugada. Esses serviços devem estar disponíveis e serem realizados em ou até determinado período de tempo ou sob determinadas condições, por exemplo.

O alto nível de disponibilidade dos serviços prestados por uma empresa de Tecnologia de Informação exige um monitoramento detalhado dos servidores envolvidos nas soluções ofertadas. A execução de rotinas em bancos de dados que são periódicas (pré-programadas) ou sob determinadas condições, planejadas ou para realizar operações que são previstas pelo sistema devem ser monitoradas. Assim, rotinas que por algum motivo, como erros do próprio sistema, não foram realizadas no tempo previsto, devem ser realizadas em outro momento.

A realização dessas rotinas pode ocorrer em período que o sistema não está sendo utilizado pelo cliente. Por exemplo: operações em banco de dados de atualização de registros que não há necessidade que sejam realizadas em tempo real, mesmo porque tomariam tempo de processamento de rotinas essenciais, como baixa de contratos, podem ser realizadas fora do horário de expediente da empresa que está usando o sistema. Outro exemplo são operações que ao serem executadas causaram erros. Nesses casos, o cliente comunica a empresa de desenvolvimento

de software que é responsável pela manutenção e está fará os procedimentos adequados para que a operação seja realizada. Esse tipo de operação precisa ser monitorado para que medidas possam ser tomadas, caso elas não sejam devidamente realizadas.

Devido à diversidade de tecnologias utilizadas na criação de soluções para esse tipo de monitoramento, nem sempre os *DataBase Administrators* (DBA) encontram no mercado ferramentas que forneçam um diagnóstico completo e adequado à realidade de cada empresa. O monitoramento da realização dessas operações que ocorrem em bancos de dados, independentemente do motivo, geram dados que podem ser utilizados pelo DBA para identificar os problemas mais recorrentes e as operações que mais causam problemas, auxiliando-o na tomada de decisão. Contribuindo, assim, para orientar a equipe de desenvolvimento na organização das rotinas a serem executadas e os arquitetos do sistema e do banco de dados no projeto e no desenvolvimento da persistência dos sistemas.

Os dados obtidos desse monitoramento podem ser utilizados em um processo de tomada de decisão que pode ser visto como *Business Intelligence* (BI), voltado para as decisões internas da equipe envolvida com as diversas atividades do ciclo de vida de software, como ênfase no desenvolvimento. De maneira genérica, BI faz uso de informações disponíveis na empresa para auxiliar os gestores (SENIOR, 2016) no processo de tomada de decisão. No contexto da proposta deste trabalho, gestor é o DBA responsável pelos sistemas cujas operações estão sendo monitoradas e em relação aos quais ele deve tomar decisões para otimizar a execução das rotinas e maximizar a taxa de sua execução com sucesso.

Considerando o conceito de BI aplicado a dados obtidos da execução de rotinas em bancos de dados de clientes, propõe-se neste trabalho o desenvolvimento de uma aplicação para monitorar a execução de processos em banco de dados de clientes. A fim de promover um monitoramento em tempo real e reduzir o tempo de tomada decisão, o software *web* desenvolvido apresentará indicadores sobre a consistência dos serviços hospedados em um conjunto de servidores. Esses dados, apresentados como gráficos e *grids* detalhados, retratarão o que ocorre com cada rotina do serviço que está sendo monitorado. Para efetuar essa verificação, cada processo que o serviço executar, será armazenado em um

banco de dados. A partir desses dados serão realizadas consultas para extrair as informações, fornecendo suporte ao processo de tomada de decisão do DBA.

## 1.2 OBJETIVOS

A seguir serão apresentados os objetivos gerais e específicos do presente trabalho.

### 1.2.1 Objetivo Geral

Desenvolver um software para monitoramento de operações realizadas em bancos de dados de clientes, disponibilizando dados sobre as rotinas executadas.

### 1.2.2 Objetivos específicos

- Realizar o levantamento das rotinas que o serviço a ser monitorado realiza.
- Produzir índices a partir dos serviços monitorados que geram resultados diferentes em termos de registros de banco de dados e *logs* de sistema operacional.
- Armazenar dados de monitoramento, indicadores, em uma base de dados de estrutura personalizada, a fim de fornecer resultados compatíveis com a natureza de cada serviço.
- Fornecer dados para suporte na identificação dos problemas mais recorrentes auxiliando o DBA na tomada de decisão.
- Monitorar operações *online* e apresentar ocorrências de problemas imediatamente após o seu acontecimento.

## 1.3 JUSTIFICATIVA

A realização deste trabalho se justifica, em termos práticos, para auxiliar DBAs na tomada de decisão sobre a melhor forma de implementação de rotinas

executadas em bancos de dados de clientes, identificação de problemas mais recorrentes nas operações que são realizadas nesses bancos. Os dados são obtidos de operações realizadas em banco de dados de clientes de sistemas produzidos por uma determinada empresa de desenvolvimento de software.

Os serviços a serem monitorados, que produzem dados utilizados para gerar informações para suporte às decisões do DBA, executam no banco de dados de cada cliente e as operações que geram esses serviços são realizadas automaticamente pelo sistema em uso pelo cliente ou sob demanda. Erros podem ocorrer na realização dessas operações. Nesse caso o registro de *logs* da realização das operações permite identificar esse tipo de ocorrência e o DBA pode ser informado imediatamente após a ocorrência do problema.

O acompanhamento da realização dessas operações permite obter dados que podem ser utilizados para auxiliar o DBA. Esses dados vistos como *logs* de operações do sistema podem auxiliá-lo em um processo de tomada de decisão em uma espécie de BI, mas voltado para a equipe de desenvolvimento.

#### 1.4 ESTRUTURA DO TRABALHO

O trabalho está organizado em 5 capítulos. No Capítulo 2 está o referencial teórico sobre BI por ser desenvolvida uma ferramenta que auxilia o DBA na tomada de decisão. O Capítulo 3 apresenta as tecnologias e as ferramentas utilizadas no desenvolvimento do trabalho. No Capítulo 4 está a apresentação do sistema desenvolvido. No Capítulo 5 estão as considerações finais.

## 2 BUSINESS INTELLIGENCE

O atual contexto dos negócios, com a globalização dos mercados e o uso de Tecnologias de Informação e Comunicação, as organizações devem ser suficientemente ágeis para antecipar-se às mudanças, encontrando soluções e adaptando-se ou criando condições para vantagens competitivas. Nesse cenário de negócios há cada vez mais a necessidade de as organizações terem disponíveis informações relevantes que as auxiliem na tomada de decisão e que tenham à disposição ferramentas para esse auxílio (MARINHEIRO; BERNARDINO, 2013). A informatização crescente das atividades de negócio e a grande concorrência têm levado as empresas à necessidade de maior controle das suas operações internas, visando melhor desempenho nos negócios pelo suporte à tomada de decisão que os dados dessas operações podem oferecer.

O *Business Intelligence* compreende diversas faces do gerenciamento, podendo envolver *Data Warehousing* e sistemas de suporte à decisão (LI, 2011). Turban et al. (2011) definem BI como um termo que combina arquitetura, ferramentas, bases de dados, aplicações e metodologias para permitir o acesso interativo a dados e a sua manipulação, fornecendo aos gerentes e analistas a habilidade de conduzir análises adequadas.

O BI surge como um conjunto de conceitos e métodos para auxiliar e aumentar a capacidade de tomada de decisão nas organizações, transformando os dados gerados pelos processos da organização em conhecimento (POWER, 2007). BI pode, também, ser entendido como o uso de diferentes fontes de informação para definir estratégias competitivas da organização (BARBIERI, 2001). BI inclui o processo de obter grandes quantidades de dados, analisá-los e produzir relatórios que sumarizam a essência das ações de um negócio que auxilia os gerentes no processo de tomada de decisão (ORACLE DATABASE, 2009). Embora BI esteja, normalmente, vinculado a grandes quantidades de dados, é importante ressaltar que BI vai além do processo de coletar grandes quantidades de dados, analisá-los e a consequente produção de relatórios que sumarizam a essência das ações no negócio (STACKOWIAK; GREENWALD, 2007).

O BI tem como principal objetivo estabelecer a conexão entre os dados gerados no dia-a-dia dos negócios das empresas, transformando-os em informação

útil, pela contextualização, categorização, organização e agregação dos dados (AMARAL; VARAJÃO, 2007). Os dados, a informação e o conhecimento compõem, assim, a vantagem competitiva para as empresas, auxiliando-as nas decisões para se diferenciarem e obter vantagem competitiva em relação à concorrência (MARINHEIRO; BERNARDINO, 2013).

Sezões, Oliveira e Baptista (2006) descrevem um conjunto de objetivos fundamentais como comuns aos sistemas de BI:

a) Acesso a dados confiáveis – a confiabilidade dos dados, a sua integração e compreensão entre diversas áreas são fundamentais para uma gestão consciente;

b) Aumento da transparência e compreensão do negócio – a disponibilização de conhecimento sobre a situação real e atual do negócio permite aos gestores e decisões uma perspectiva das áreas que devem controlar e aumentar a sua capacidade de entendimento do negócio;

c) Suporte para a tomada de decisão – o conhecimento produzido pelos sistemas de BI, potencializados pelas tecnologias de comunicação, deve suportar e justificar as medidas tomadas pelos vários intervenientes no processo de gestão.

Ferramentas de BI são vistas como tecnologias que permitem a eficiência de operações de negócio por providenciar um valor agregado à informação da empresa auxiliando no processo de tomada de decisão (MILLER; MCROBBIE; BEGG, 2016). A visão gerencial do BI é fornecer a informação correta para a pessoa certa no tempo certo de maneira que ela possa tomar decisões que em última análise melhoram o desempenho organizacional (BOSE, 2009). Crucialmente todas as organizações precisam melhorar continuamente seu desempenho para permanecerem competitivas e, assim, o uso correto e continuado de BI é considerado uma maneira de alcançar essa melhoria (MILLER; MCROBBIE; BEGG, 2016).

Há um número crescente de empresas que fornecem BI para os tomadores de decisão (internos e externos) (GUARDA et al., 2016). Internamente, o suporte de BI, reforça a responsabilidade de todos os colaboradores e melhora a estabilidade da gestão. E externamente, relacionamentos com fornecedores e parceiros de negócio podem ser reforçados por meio de um efetivo compartilhamento de indicadores de desempenho para benefícios mútuos (XIE et al., 2007).



É muito importante, embora possa ser bastante difícil, para a organização tomar decisões acertadas. Ter a habilidade de tomar as decisões corretas é essencial para aumento de lucros, gerenciamento de riscos e o bom desempenho global da empresa. Devido aos fatores não controláveis tais como os mercados em movimento rápido, as mudanças da economia e da legislação e os novos competidores (frutos, inclusive, da globalização), tomar as decisões corretas não é uma tarefa trivial (GUARDA et al., 2016). O uso de tecnologias pode auxiliar nesse processo. Elas permitem armazenar dados das operações de negócio e analisar esses dados para informações que podem ser úteis para o tomador de decisões.

Guarda et al. (2016) consideram BI como o processo por meio do qual os usuários obtêm dados acurados e consistentes a partir dos dados armazenados pela organização. Os dados, obtidos a partir de vários contextos de negócio, permitem identificar, analisar e detectar tendências, oportunidades, ameaças e anomalias e, assim, serem usados para a tomada de decisão. Sistemas e ferramentas de BI desempenham um papel relevante no processo de planejamento estratégico da organização. Esses sistemas permitem coletar, armazenar, acessar e analisar dados a fim de suportar e facilitar o processo de tomada de decisão (AAKER, 2009).

As organizações desenvolvem suas estratégias para manter ou obter vantagens competitivas, sendo, assim, reféns das ferramentas e sistemas de BI (GUARDA et al., 2016). A dependência desses sistemas e ferramentas decorre da quantidade de dados que são gerados pelos processos de negócio e que as empresas armazenam e pela complexidade das ações e decisões envolvidas nesses negócios. Essas são decorrentes, também, da abertura de mercados e de clientes mais exigentes e seletivos pela oferta e acesso à informação e pelo aumento da concorrência.

### 3 MATERIAIS E MÉTODO

Neste capítulo estão descritas as tecnologias e as ferramentas utilizadas na modelagem e na implementação do sistema.

#### 3.1 MATERIAIS

O Quadro 1 apresenta os materiais e as ferramentas utilizados na modelagem e na implementação do sistema.

Ferramenta / Tecnologia	Versão	Referência	Finalidade
Visual Studio Community	2015	<a href="https://www.visualstudio.com/pt-br/vs/community/">https://www.visualstudio.com/pt-br/vs/community/</a>	Ambiente de Desenvolvimento Integrado.
ASP.NET/.NET Framework	4.5	<a href="https://www.microsoft.com/net/download/framework">https://www.microsoft.com/net/download/framework</a>	Ambiente de execução / <i>framework</i> de desenvolvimento.
Visual Basic		<a href="https://msdn.microsoft.com/pt-BR/library/2x7h1hfk.aspx">https://msdn.microsoft.com/pt-BR/library/2x7h1hfk.aspx</a>	Linguagem de programação.
Case Studio 2	2.25.0	<a href="http://www.casestudio.com/enu/index.aspx">http://www.casestudio.com/enu/index.aspx</a>	Criação da modelagem do banco de dados (Diagrama de entidade e relacionamento).
HTML5		<a href="https://developer.mozilla.org/pt-BR/docs/Web/HTML/HTML5">https://developer.mozilla.org/pt-BR/docs/Web/HTML/HTML5</a>	Linguagem para estruturação e apresentação de conteúdo.
Bootstrap	3.3.7	<a href="http://getbootstrap.com/">http://getbootstrap.com/</a>	Desenvolvimento front-end.
Microsoft SQL Server 2014	12.0.200.8	<a href="https://pt.wikipedia.org/wiki/Microsoft_SQL_Server">https://pt.wikipedia.org/wiki/Microsoft_SQL_Server</a>	Sistema gerenciador de Banco de dados relacional.
Microsoft SQL Server Management Studio	13.0.16106.4	<a href="https://msdn.microsoft.com/pt-BR/library/mt238290.aspx">https://msdn.microsoft.com/pt-BR/library/mt238290.aspx</a>	Ambiente integrado para acessar, configurar, gerenciar, administrar e desenvolver todos os componentes do SQL Server.
Ajax		<a href="http://codigofonte.uol.com.br/artigos/o-que-e-o-ajax-e-como-ele-funciona">http://codigofonte.uol.com.br/artigos/o-que-e-o-ajax-e-como-ele-funciona</a>	Carregar e renderizar página, utilizando recursos de scripts executando no cliente.
Jquery		<a href="https://jquery.com/">https://jquery.com/</a>	JQuery é uma biblioteca JavaScript..
Java Script		<a href="https://www.javascript.com/">https://www.javascript.com/</a>	É uma linguagem de programação interpretada.
CSS3		<a href="http://www.css3.info/">http://www.css3.info/</a>	Para definição de estilo para as páginas web com efeitos de transição, imagem e outros.
Balsamiq Mockups	2.6.0.4	<a href="https://balsamiq.com/">https://balsamiq.com/</a>	Criador de protótipos de interface do sistema.
IIS	10.0.14393.0	<a href="https://msdn.microsoft.com/pt-br/library/hh831725(v=ws.11).aspx">https://msdn.microsoft.com/pt-br/library/hh831725(v=ws.11).aspx</a>	Servidor web para hospedagem do sistema desenvolvido.

**Quadro 1 – Ferramentas e tecnologias utilizadas**

A seguir são apresentados os materiais e as ferramentas utilizadas para o desenvolvimento do aplicativo.

### 3.1.1 Visual Studio 2015

O Microsoft Visual Studio é um pacote de ferramentas, componentes e tecnologias de desenvolvimento de software da Microsoft, especialmente dedicado ao *framework.NET* e às linguagens *Visual Basic* (VB), C, C++, C Sharp (C#) e J Sharp (J#). O Visual Studio é utilizado para desenvolvimento de sistemas *desktop*, *web* e para dispositivos móveis, com a plataforma do *ASP.NET* e serviços *web XML*. As linguagens mais utilizadas nessa plataforma são: Visual Basic .Net (VB.NET) e C#.

Com o uso do Visual Studio é possível simplificar o processo de desenvolvimento, depuração e implantação de aplicativos Microsoft, pois este Ambiente Integrado de Desenvolvimento, ou *Integrated Development Environment* (IDE), conta com integração com recursos como *SharePoint* e computação em nuvem (VISUAL STUDIO, 2016).

Além disto, esse ambiente possui suporte integrado para desenvolvimento orientado a testes com ferramentas de depuração (*debug*) e de *design* e possibilita a adição de *plugins* que permitem expandir suas funcionalidades. O Visual Studio é compatível com os bancos de dados *Microsoft SQL Server*, *Oracle* e *IBM DB2* (VISUAL STUDIO, 2016).

### 3.1.2 Microsoft Visual Basic.Net

O Visual Basic (VB) é uma linguagem de programação que proporciona uma IDE que torna a criação e o desenvolvimento de interfaces e aplicações mais simples. Popularmente chamada de VB, a linguagem faz parte do pacote Microsoft Visual Studio.

Lançado no final de 2001 o Visual Basic.Net chamado VB.NET é uma linguagem de programação orientada a objetos, que pode ser considerada como

uma evolução do Visual Basic. Dispõe de novas possibilidades como o desenvolvimento de aplicações para a *web*, ampliando o Desenvolvimento Rápido de Aplicação (*Rapid Application Development - RAD*), que já existia no Windows, para a rede com aplicações mais escaláveis e robustas (VB NET, 2017).

O VB possibilita desenvolver aplicativos para Windows, *web* e dispositivos móveis. A partir de 2000, com o lançamento do Visual Studio.NET baseado na plataforma .NET, a Microsoft optou por um novo paradigma de programação para o VB, surgindo o VB.NET como uma das linguagens integrantes do pacote Visual Studio .NET, junto com C#. O Quick Basic, que é o núcleo da linguagem, permaneceu o mesmo.

### 3.1.3 Asp Net

ASP.NET é um *framework web* livre, utilizado na construção de sites baseados em padrões com *HyperText Markup Language 5 (HTML5)*, *Cascading Style Sheets (CSS)* e JavaScript, suporta três tipos de abordagens para construção de sites: o Web Forms, ASP.NET MVC e Web ASP.NET. Os formulários ou páginas ASP.NET possuem extensão “.aspx” e os arquivos de código das páginas, possuem extensão “.vb”, no caso de a linguagem ser Visual Basic (ASP NET, 2017).

### 3.1.4 Net Framework

O .Net Framework é uma plataforma de desenvolvimento para a criação de aplicativos para Windows, Windows Phone, Windows Store, Windows Server e Windows Azure. A plataforma .Net Framework inclui as linguagens de programação C# e Visual Basic e uma ampla biblioteca de classes. Essa plataforma oferece um ambiente de execução gerenciado de desenvolvimento e implantação simplificados e integração com uma variedade de linguagens de programação, como o Visual Basic e o Visual C#.

O .Net Framework é uma tecnologia que oferece suporte à criação e à execução de aplicativos e serviços *eXtensible Markup Language (XML)* da *web*, foi

criado para fornecer um ambiente de programação orientada a objetos consistente. E fornece outras vantagens como (MSDN MICROSOFT, 2017):

a) Ambiente de execução que minimiza conflitos de versionamento de publicação.

b) Ambiente de execução que elimina os problemas de desempenho dos ambientes interpretados ou com *scripts*.

c) Executa a comunicação usando padrões, garantindo que códigos baseados no .Net Framework possam integrar-se a outros códigos.

### 3.1.5 Case Studio

Ferramentas *Computer Aided Software Engineering* (CASE) são comumente utilizadas na Engenharia de Software. Essas ferramentas auxiliam os analistas na construção de estruturas de dados, como as principais classes e entidades de um sistema e os relacionamentos entre elas.

O Case Studio é uma ferramenta profissional para projeto de banco de dados, para geração de estruturas de banco de dados, tabelas, *scripts* SQL, relatórios *HTML* entre outros (CASE STUDIO, 2017).

O Case Studio cria visualmente Diagramas de Entidade Relacionamento (DER) em vários bancos de dados. Possui recursos como implementação de relacionamentos de forma automática e controle de usuários, além disso, possibilita um bom nível de abstração dos diagramas. Suporta a geração de *scripts* dos principais bancos de dados atuais como *MySql*, *PostgreSQL*, *Oracle*, *Access*, *Interbase* e *Firebird*, entre outros (YAMAZACK, 2017).

### 3.1.6 HTML 5

*Hyper Text Markup Language*, em português linguagem para marcação de hipertexto, está na sétima versão, passando por: HTML, HTML+, HTML 2.0, HTML 3.0, HTML 4.0, HTML 4.0.1 e HTML 5.

As especificações para a HTML 5 estão estruturadas em 10 seções que orientam os desenvolvedores no que tange a compatibilidade, utilidade, interoperabilidade e acesso universal, sendo elas (SILVA, 2016, p.27):

- a) Infraestrutura – terminologias, classes e sintaxes.
- b) Semântica, estrutura e interface de programação de aplicativos para documentos HTML – funcionalidades e elementos do HTML.
- c) Elementos HTML – regras de uso dos elementos, marcação e diretrizes.
- d) Micro formatos – mecanismos capazes de converter HTML em outros formatos para serem lidos por computadores.
- e) Carregamento de páginas web – tratamento de páginas web em diferentes dispositivos.
- f) APIs para aplicações web – desenvolvimento de *scripts* em aplicações HTML.
- g) Interações com o usuário – mecanismos de interação em documentos HTML.
- h) APIs para comunicação – comunicação entre aplicações executando em domínios diferentes.

Dentre as vantagens da HTML 5 está a introdução de novos elementos de marcação, melhores opções de estruturação de documentos, exibição e reprodução de conteúdo de áudio e vídeo. Além disso, a funcionalidade de geolocalização, também ganhou notoriedade com o uso da HTML 5 e tem como característica trazer informações relacionados a localização geográfica do dispositivo (SILVA, 2016, p.215).

### **3.1.7 Bootstrap**

O Bootstrap tem por objetivo facilitar o desenvolvimento de interfaces (*front-end*) para páginas web. Esse *framework* tem padrões para elementos HTML comuns ao usuário, além de elementos personalizados como classes CSS padrão.

O Bootstrap é um *framework* de desenvolvimento utilizado para criar aplicações web, *desktop* e *mobile* utilizando HTML, CSS e JavaScript, contém diversos componentes como botões, campos de entrada, *labels*, listas, iconografia e

emblemas, entre outros. Fornece algumas funcionalidades específicas como transição, caixas de diálogo, *tooltips* e alertas, além de *plugins* do jQuery.

O Bootstrap começou como um projeto interno do Twitter, mas logo se tornou um projeto *open source*, sendo um dos mais ativos no GitHub com mais de 300 contribuintes (AVRAM, 2017).

O Bootstrap possui diversos componentes que melhoram a produtividade do desenvolvedor. Inicialmente foi desenvolvido para navegadores mais atuais, mas possui compatibilidade com navegadores mais antigos. Adota o conceito de *web design* responsivo que adapta sua exibição para o ambiente de visualização, também possui uma excelente documentação e componentes funcionais, além dos *plugins* em jQuery (MARQUES, 2017).

### **3.1.8 SQL Server 2014 Express**

O Microsoft SQL Server ou SQL Server Express é um aplicativo para gerenciamento de bancos de dados que dispõe de recursos de segurança de dados, além de desempenho para aplicações cliente/servidor, aplicações *web* e armazenamento local. O Microsoft SQL Server 2012 Express tem distribuição gratuita e fornece um repositório de dados confiável e avançado para aplicativos *desktop* e *web*, podendo ser redistribuído e integrado com outras aplicações.

Criado para uma implantação facilitada e protótipos rápidos, com ele é possível criar aplicações robustas, usando recursos avançados e também centralizar e aumentar o desempenho do aplicativo com procedimentos armazenados de integração, *triggers*, e *views*. Conta com suporte para dados estruturados e não estruturados, dados relacionais, como a XML e dados espaciais. Adiciona informações geográficas em aplicações, armazena e valida dados XML e XQuery. Permite interoperabilidade do SQL Server Express, usando o driver SQL Server para Hypertext Preprocessor (PHP), e o driver JDBC da Microsoft para a implementação Java (SQL SERVER, 2017).

### **3.1.9 Microsoft SQL Server Management Studio Express**

O Microsoft SQL Server Management Studio Express é um ambiente de desenvolvimento integrado gratuito utilizado para configurar, gerenciar, administrar e desenvolver os componentes do SQL Server, com base em uma interface gráfica. Disponibiliza um amplo grupo de ferramentas gráficas e editores de *script*, que permitem a integração com o SQL Server (MANAGEMENT STUDIO, 2017).

### **3.1.10 Ajax**

Ajax significa Asynchronous JavaScript and XML, em sua essência, é uma forma eficiente de uma aplicação *web* tratar as interações de um usuário com uma página *web*. Ele reduz a necessidade de fazer uma atualização da página ou uma recarga completa da página a cada interação do usuário.

Isso permite que a aplicação tenha um comportamento semelhante ao de uma aplicação *desktop* com as interações do Ajax sendo tratadas assincronamente em segundo plano.

As interações do Ajax são iniciadas pelo código JavaScript e quando a interação é concluída, JavaScript atualiza o código-fonte HTML da página. Desta forma as alterações são feitas imediatamente sem requerer uma atualização da página. As interações podem ser utilizadas para validar entradas do formulário, por exemplo, enquanto um usuário está digitando, utilizar a lógica do lado do servidor, recuperar dados detalhados do servidor ou atualizar dinamicamente os dados em uma página (NET BEANS, 2017).

Um exemplo simples de aplicação do Ajax é o sistema de conversas do Gmail, webmail do Google, com uma aba com a indicação mensagem inicial é exibida, porém o seu conteúdo fica oculto. Ao clicar sobre esta aba, ela exhibe automaticamente o texto correspondente e o oculta novamente caso o usuário deseje, sem que seja necessário recarregar toda a página (MOREIRA, 2017).



### 3.1.11 JQuery

jQuery é uma biblioteca JavaScript *cross-browser* desenvolvida para simplificar os *scripts*, veio ajudar a resolver problemas de incompatibilidades entre os navegadores, reduzir o tamanho de código e introduzir a reutilização de código através de *plugins*. Além disso, permite a implementação de recursos de CSS e trabalha com *Ajax* e *DOM*.

jQuery é a mais popular das bibliotecas *JavaScript*, faz uso da licença *Massachusetts Institute of Technology* (MIT) ou da *General Public License* (GNU). A sintaxe do *jQuery* foi desenvolvida para tornar mais simples a navegação do documento HTML e a seleção de elementos DOM, também é usada para criar animações, manipular eventos e desenvolver aplicações *Ajax*, esta biblioteca oferece a possibilidade de criação de *plugins* (JQUERY, 2017).

### 3.1.12 JavaScript

*JavaScript* é uma linguagem de programação do lado do cliente da aplicação (ALVAREZ, 2017). Como se tornou compatível com a maioria dos navegadores é a linguagem de programação do lado do cliente mais utilizada atualmente (JAVASCRIPT, 2017).

*JavaScript* é uma linguagem de programação consideravelmente leve, interpretada e com recursos de orientação a objetos, permite que um conteúdo executável seja incluído em páginas *web*. Dessa forma, uma página *web* não precisa mais de HTML estático e pode incluir programas que interagem com o usuário, controlam o navegador e criam conteúdo HTML dinamicamente. Seu objetivo é atender as limitações apresentadas pelo HTML (FLANAGAN, 2002).

### 3.1.13 Cascading Style Sheet 3

O *Cascading Style Sheets 3* é responsável pelo aspecto visual das páginas *web*, pelo layout, garante acessibilidade, tanto para aplicativos computacionais quando para as pessoas, agiliza a leitura das páginas *web* pelos navegadores. Não é uma linguagem de programação nem de marcação, sendo uma sequência de

declarações de propriedades e seus valores para a manipulação da forma como o conteúdo de uma página *web* será exibido (GOMES, 2010).

A grande vantagem do uso de CSS é a de separar a marcação HTML, da apresentação da página *web*. HTML marca e estrutura textos, cabeçalhos, parágrafos, *links*, botões, formulários, imagens e demais elementos da página e CSS define cores, posicionamento na tela, estilos de linhas, bordas e tudo o mais relacionado à apresentação (SILVA, 2016).

### **3.1.14 Bizagi Process Modeler**

O *Bizagi Process Modeler* é um software utilizado para representar graficamente todos os processos de negócio com uma modelagem descritiva, analítica e de execução. É, também, muito utilizado a fim de descrever fluxos de trabalhos para melhoria de tempo de execução e custo das atividades que foram desenvolvidas (BIZAGI, 2017). O Bizagi utiliza a notação de modelagem *Business Process Management Notation* (BPMN).

### **3.1.15 Internet Information Services**

*Internet Information Services* (IIS) é o servidor *web* da Microsoft. Fornece uma plataforma *web* segura, modular e simples de gerenciar a hospedagem de *sites*, serviços e aplicativos. O IIS é uma plataforma unificada que integra IIS, ASP.NET, serviços de *File Transfer Protocol* (FTP), PHP Hypertext Preprocessor (PHP) e *Windows Communication Foundation* (WCF). Com o IIS é possível realizar o isolamento automático de aplicações, programar vários tipos de aplicações no mesmo servidor e diversas outras configurações como áreas restritas, reduzindo riscos de segurança (MSDN MICROSOFT, 2016).

IIS poderá vir habilitado com o sistema operacional Windows ou ser habilitado nas ferramentas e recursos, tanto para as máquinas de desenvolvimento, quanto para o servidor.

### 3.1.16 Balsamiq Mokuups

O *Balsamiq* é um software muito importante para o desenvolvimento de sistemas, ele permite que o profissional (desenvolvedor) explicar ao usuário a ideia do desenvolvimento por protótipos de telas de forma rápida e simples de fazer. (BALSAMIQ, 2017).

## 3.2 MÉTODO

Antes do processo de desenvolvimento de software foi realizada uma pesquisa bibliográfica das ferramentas a serem utilizadas, visando obter informações sobre os principais recursos disponíveis em cada ferramenta, para posterior utilização desses recursos durante o desenvolvimento do projeto.

O desenvolvimento do trabalho foi dividido em algumas fases. Essas fases estão descritas nos itens a seguir:

### a) Levantamento de requisitos

Nessa fase foram levantadas as dificuldades e as necessidades encontradas no dia-a-dia da empresa para a qual o aplicativo foi desenvolvido. Essas dificuldades estavam relacionadas a saber se todos os serviços e suas respectivas rotinas estavam sendo executadas com êxito. Atualmente nessa empresa existem mais de 25 serviços em execução, alguns simultaneamente e outros em horários específicos. Contudo, notou-se uma necessidade muito grande de acompanhar se todas as rotinas estabelecidas estão sendo executadas efetivamente e se ocorre algum erro.

Foi elaborado um esboço inicial do padrão das telas para o sistema e dos requisitos para o plano de testes.

### b) Análise

Nessa etapa foram analisados os fatores, as necessidades e o objetivo final do aplicativo, visando desenvolver um projeto mais simples possível e trazendo as informações com rapidez, para assim alcançar a solução esperada.

### c) Projeto

Nessa fase foi definida a forma de desenvolvimento do sistema. Incluindo a escolha das ferramentas e das tecnologias que posteriormente auxiliariam em praticamente todo o processo de desenvolvimento.

#### **d) Implementação do sistema**

Nessa etapa foram analisadas várias formas de implementação e desenvolvimento que atendessem da maneira mais prática e objetiva e que auxiliassem, também, na manutenção do sistema.

Foi escolhido o Visual Studio como ferramenta para codificação do sistema. A linguagem utilizada foi o Visual Basic.Net. A versão do Net Framework utilizada é a 4.5, para o *front-end* do sistema foi utilizado o Bootstrap devido a sua facilidade de implementação dos recursos visuais necessários. Devido à familiaridade do autor deste trabalho, foi optado pela utilização do SQL Server como Sistema Gerenciado de banco de dados (SGBD) do sistema e como sistema de gerenciamento do SGBD foi utilizar o *SQL Server Management Studio (SSMS)* já que eles são integrados.

Como o sistema vai possuir gráficos que serão os principais responsáveis para apresentar ao usuário (DBA) as informações para tomada de decisão, notou-se a necessidade da utilização de JQuery e JavaScript. O DBA também poderá verificar dados mais detalhados do serviço monitorado, por meio de um *highlight* chamado via HTML5. Para alguns componentes de filtro e listagem foi optado por utilizar o Ajax devido ao seu uso anterior em outros sistemas. E, por fim, a hospedagem do sistema foi realizada no IIS.

A implementação do sistema iniciou-se com a instalação do Visual Studio 2015. O instalador pode ser *online* ou *off-line*. Além de desenvolvimento em outros dispositivos e também outras linguagens. A linguagem VB.Net, que foi utilizada no desenvolvimento já vem configurada por padrão.

Para realizar o desenvolvimento, é necessário habilitar o IIS, nos componentes do *Windows*. Em alguns casos, as opções necessárias já vêm marcadas, porém é importante verificar. Na Figura 1 após acessar a opção ativar ou desativar recursos do *Windows*, verifica-se os itens que deverão ficar marcados (habilitados).

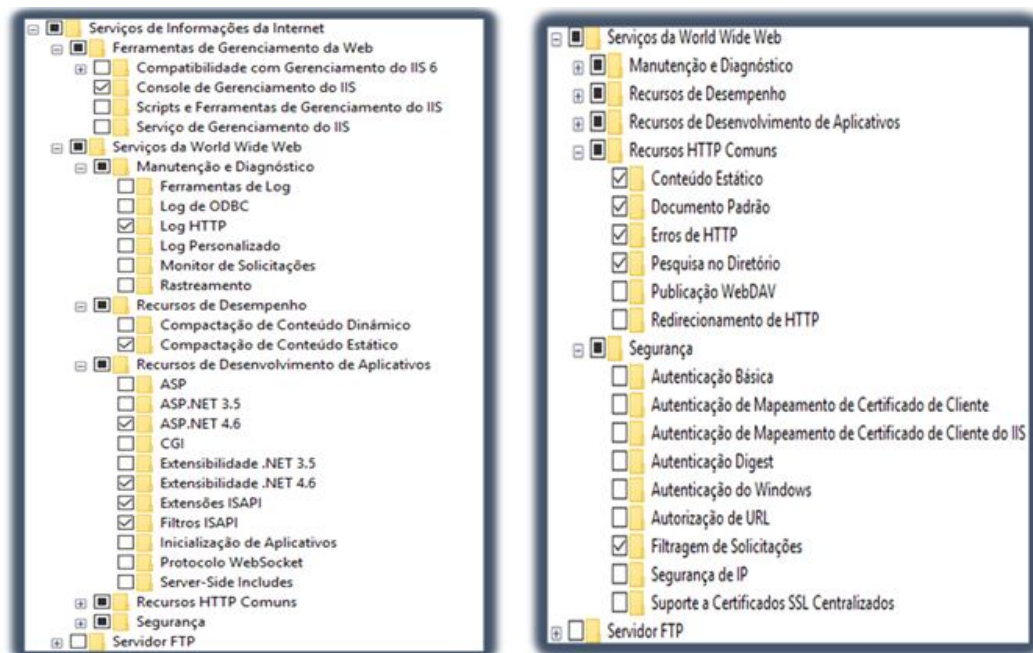


Figura 1 – Habilitar o IIS

É possível executar o sistema utilizando somente o Visual Studio como servidor na máquina de desenvolvimento. Entretanto, utilizar o IIS é mais interessante, pois é possível simular a hospedagem do sistema e suas configurações. Na Figura 2 é exibido o Visual Studio 2015, com o projeto do sistema em execução.

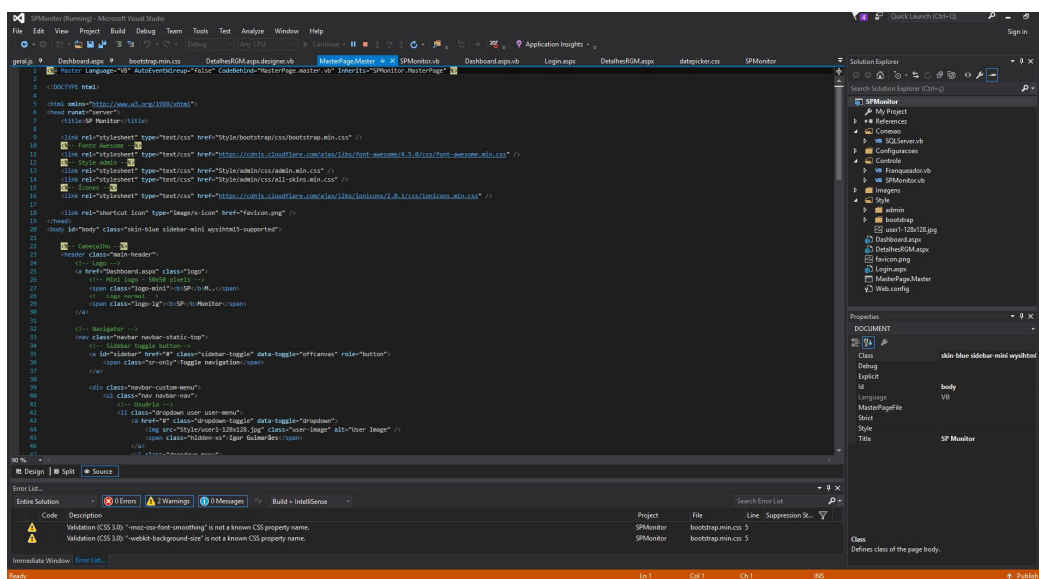


Figura 2 – Visual Studio 2015

### **e) Realização dos testes**

Enquanto estava sendo desenvolvido o processo de codificação, simultaneamente estavam sendo realizados os testes operacionais, visando representar as ações do usuário final e procurando simular as mais diversas formas de operações a serem realizadas. Foi possível prever algumas tarefas a serem executadas e desenvolver um tratamento no código nas possíveis exceções. Por fim, foram executados mais uma série de testes com o sistema já compilado, com o objetivo de simular a real situação em que o sistema se encontrará após implementado.

## 4 RESULTADO

Este Capítulo apresenta o *software* de monitoramento, detalhando a análise do sistema, a tela deste e recursos de códigos fontes utilizados. Na seção 4.1 é encontrada a descrição do sistema, o que ele faz, seus requisitos e restrições.

### 4.1 ESCOPO DO SISTEMA

O sistema visa possibilitar monitoramento em tempo real e reduzir o tempo de tomada decisão de DBAs. O sistema *web* apresentará indicadores sobre a consistência dos serviços hospedados em um conjunto de servidores de forma personalizada. Para isso haverá:

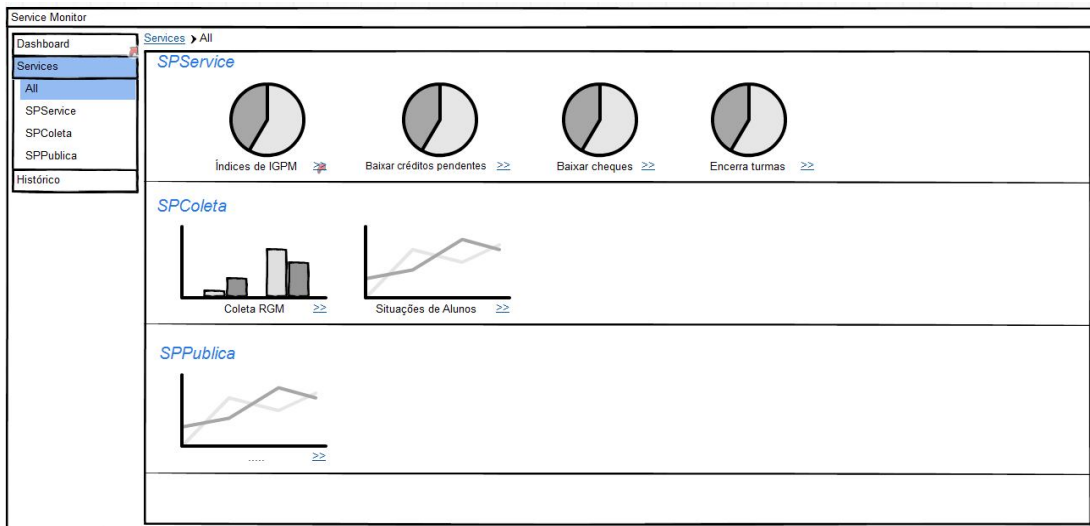
1) A produção de índices a partir de cada um dos serviços monitorados. Cada serviço produzirá um resultado diferente em quesitos como registros de banco de dados e *logs* de sistema operacional.

2) Armazenamento de dados na base utilizada pela ferramenta. Os indicadores produzidos serão armazenados em uma base de dados de estrutura personalizada, a fim de fornecer resultados compatíveis com a natureza de cada serviço.

3) Desenvolvimento *front-end* para consumo das informações. O *front-end* da aplicação será composto por componentes que exibam índices e gráficos em tempo real, fornecendo uma visualização consolidada ao administrador do sistema.

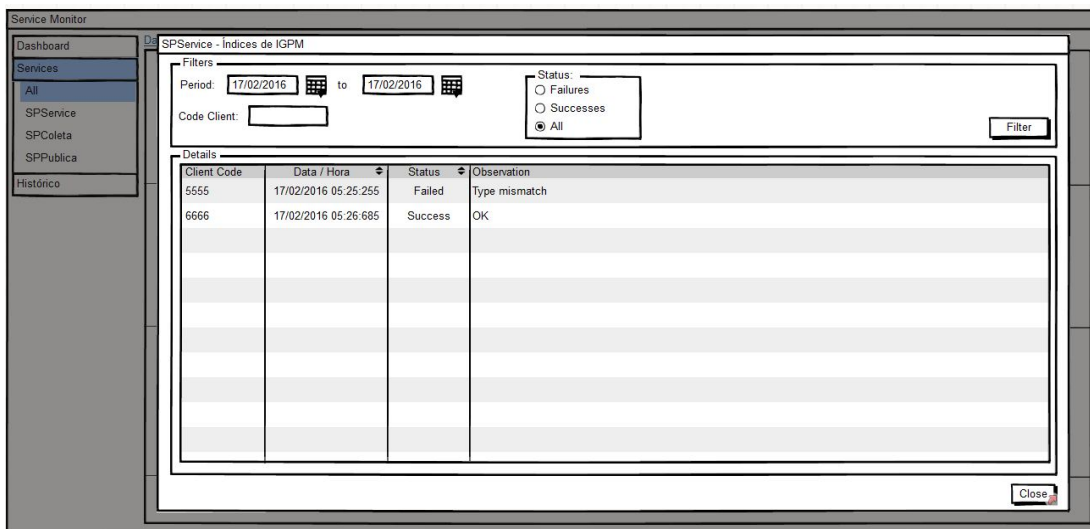
### 4.2 MODELAGEM DO SISTEMA

Nesta seção será descrito o processo de negócio do sistema, com as respectivas ações de cada participante. Na Figura 3 está o um *mockup* do *dashboard* do sistema, utilizando o Balsamiq, *que foi criado para* fornecer uma visão geral de como os serviços e os índices que estão sendo monitorados são apresentados para o usuário.



**Figura 3 – Mockup: dashboard da aplicação**

Na Figura 4 está o *mockups* criado para os detalhes do serviço selecionado, contendo filtros para busca dos índices que são apresentados no *grid*.



**Figura 4 – Mockup: detalhes do serviço**

Utilizando o Bizagi foi criado um diagrama para apresentar todo o processo do sistema. Esse diagrama de processo de negócio (semelhante a um diagrama de atividades) apresenta as principais atividades de processo realizadas e o fluxo de integração entre elas. A Figura 5 apresenta o processo de negócio da aplicação. A notação adotada é a definida por BPMN.



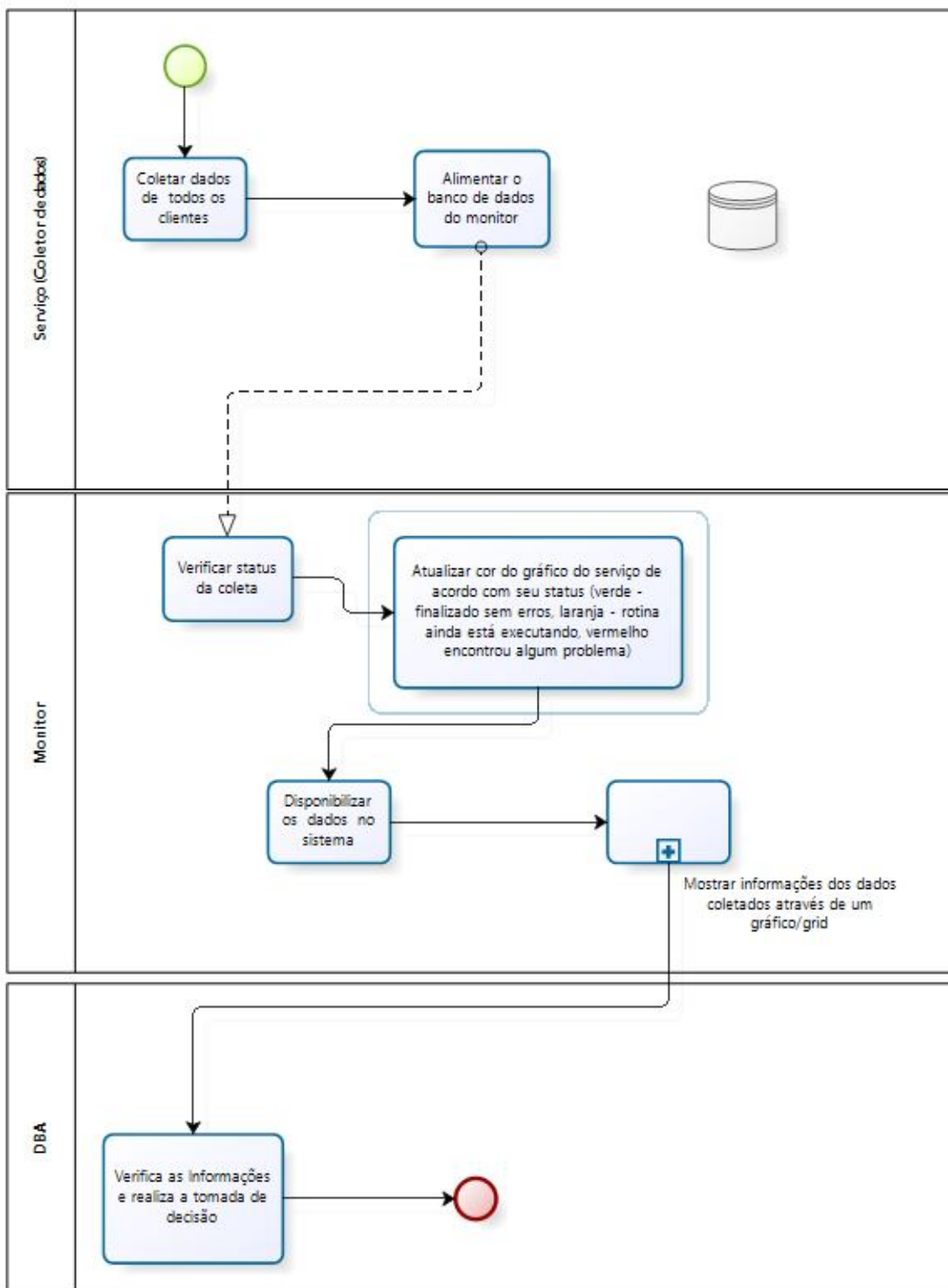


Figura 5 – Diagrama: processo do sistema

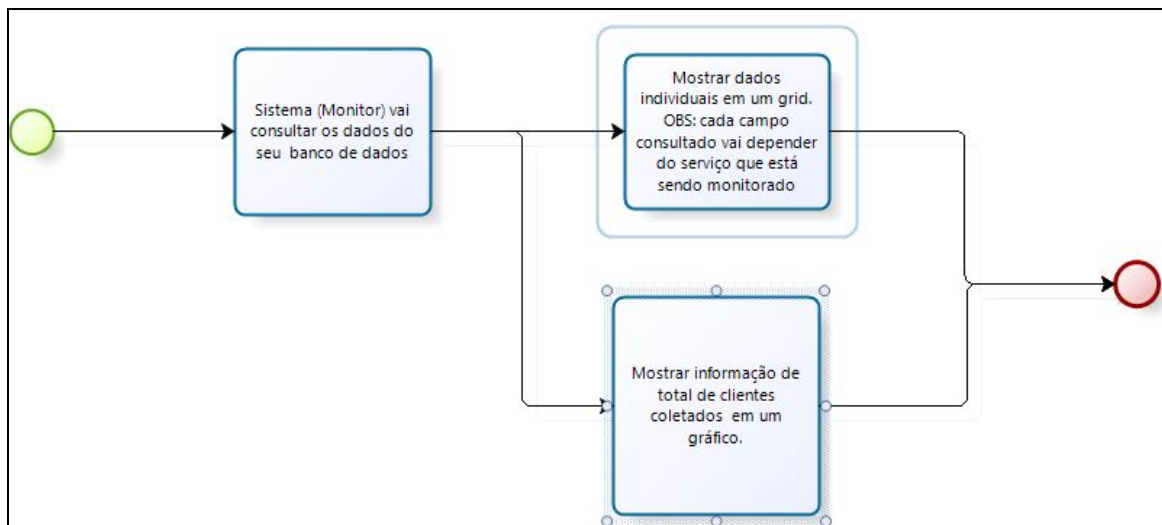
Na Figura 6 são apresentados todos os indivíduos envolvidos no uso do sistema (retângulos verticais à esquerda) e o que cada um é responsável por fazer:

a) serviço (coletor de dados) - fornecerá os índices coletados no banco de dados. Os serviços poderão ter índices diferentes entre si;

b) monitor (sistema) - será o responsável por manipular as informações algumas informações e disponibilizá-las em tela;

c) DBA - verificará todas as informações que o sistema apresentar e realizará tomada de decisão.

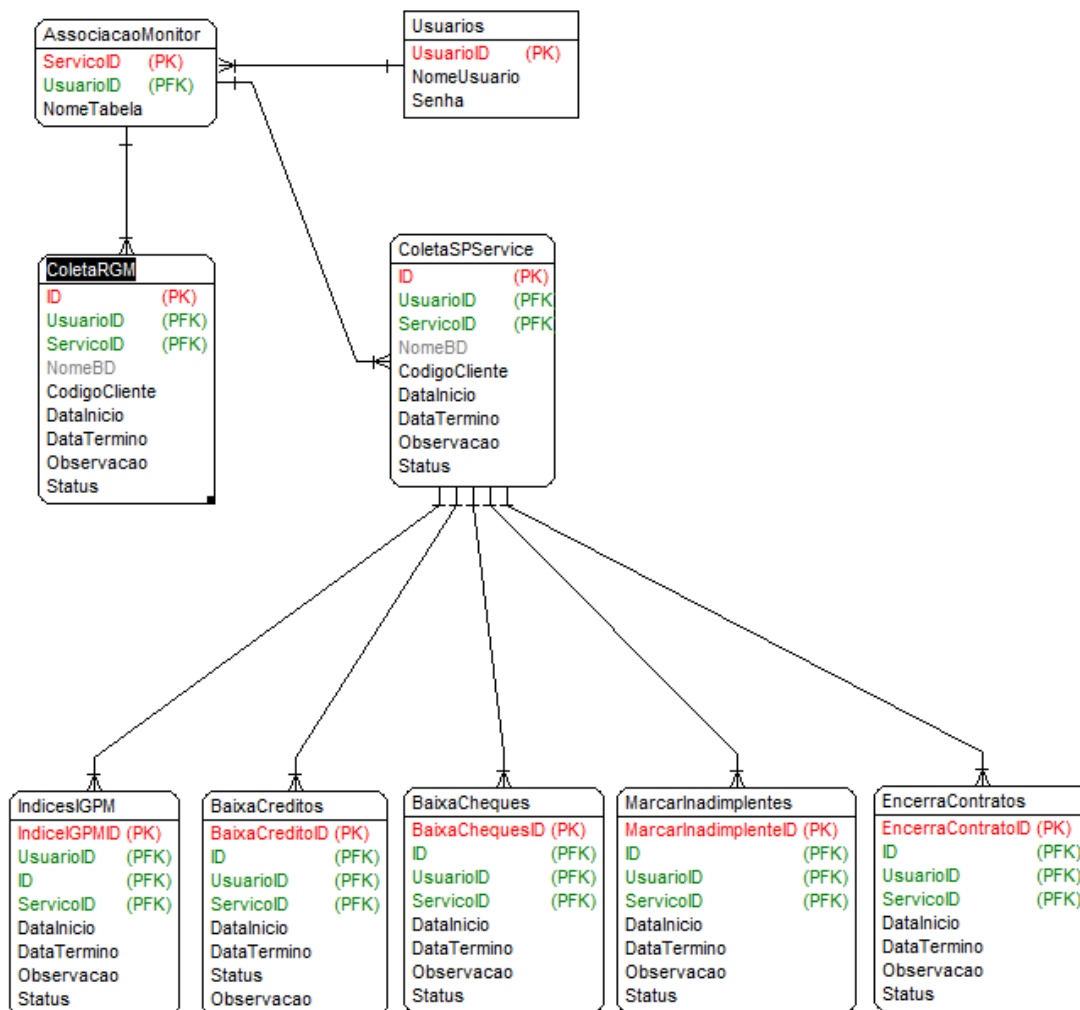
Na Figura 4 está o final do processo do sistema (monitor). Após obter as informações necessárias disponibilizará os dados visualmente ao usuário (DBA) por meio de gráfico e mais detalhadamente em um *grid* com os detalhes do serviço previamente selecionado.



**Figura 6 – Diagrama: processo do sistema detalhes monitor**

Para a modelagem de dados foi utilizado o Case Studio, que permite ao desenvolvedor criar o modelo visual das tabelas. Esse diagrama criado é apresentado na Figura 7. O Case Studio possibilita nomear os campos, definir atributos desses campos e o tipo de relacionamento entre as entidades do banco (as tabelas).

O próprio Case Studio realiza a exportação do *script* para criação do banco de dados no *SQL Server 2014*.



**Figura 7 – Diagrama de Entidade e Relacionamento**

Na Figura 7 é apresentando o *DER* do banco de dados. Cada nova tabela de serviço de coleta de dados deverá receber chaves estrangeiras das tabelas de cliente e associacaomonitor. Para manter um padrão de desenvolvimento e modelagem, foram utilizadas as datas de início e de término para as sub-rotinas do serviço para ser possível definir o tempo de execução de cada sub-rotina do serviço e do próprio serviço como um todo. A coleta dos dados pode ser feita diretamente dentro da rotina do serviço (via código) ou via *trigger* no banco de dados dos clientes no qual a rotina é executada. No desenvolvimento deste trabalho, optou-se por fazer direto na rotina do serviço por ser mais rápido para desenvolver e ter acesso ao código da rotina.

Cada serviço monitorado terá suas próprias tabelas individuais devido ao fato de cada serviço possuir índices diferentes para coleta. A estrutura apresentada na Figura 7 foi criada para os serviços de Relatório Gerencial Mensal (RGM) e SPService, como o RGM possui apenas uma rotina em seu serviço, foi criado apenas uma tabela para armazenar seus índices. O SPService possui 5 rotinas por isso foram criadas 6 tabelas, uma principal que retornará os índices gerais de todas as rotinas e 5 tabelas individuais para cada rotina que ele executa. A tabela de usuários e associacaomonitor deverá ser associada a qualquer serviço novo que venha a ser monitorado, pois é nela que será possível identificar a qual tabela os serviços em questões fornecerão os dados para coleta.

#### 4.3 DESCRIÇÃO DO SISTEMA

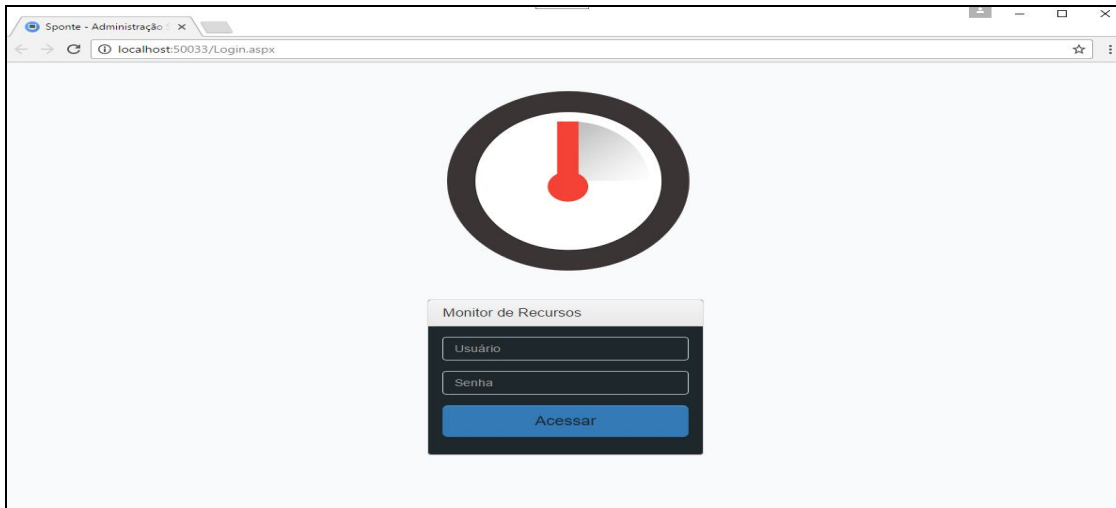
O sistema desenvolvido possibilita monitoramento em tempo real visando reduzir o tempo de tomada decisão de DBAs. O sistema, que é *web*, apresenta indicadores sobre a consistência dos serviços hospedados em um conjunto de servidores personalizados. Para isso haverá:

1) A produção de índices a partir de cada um dos serviços monitorados. Cada serviço produzirá um resultado diferente em relação aos registros de banco de dados e *logs* de sistema operacional.

2) Armazenamento de dados na base utilizada pela ferramenta. Os indicadores produzidos serão armazenados em uma base de dados de estrutura personalizada, a fim de fornecer resultados compatíveis com a natureza de cada serviço.

3) Desenvolvimento *front-end* para consumo das informações. O *front-end* da aplicação é composto por componentes que exibem índices e gráficos, dependendo do serviço em tempo real ou não, fornecendo uma visualização consolidada para o administrador do sistema.

A seguir serão apresentadas telas do sistema, sendo imagens obtidas por *print screen* do próprio sistema. Na Figura 8 é apresentada a tela de *login* do sistema. Somente usuário previamente cadastrado no banco de dados do sistema poderá ter acesso.



**Figura 8 – Tela de login do sistema**

Na Figura 9 está a tela que é apresentada após o usuário acessar o sistema. Nessa tela são apresentados os menus (área lateral esquerda), um *dashboard* com os gráficos dos sistemas que estão sendo monitorados e no canto superior direito é apresentado o nome do usuário que está logado.

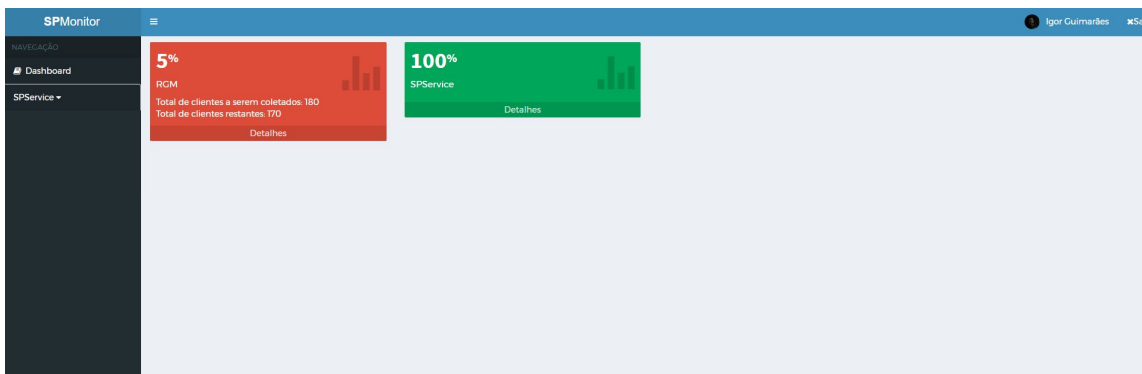
O *status* dos gráficos apresentados é representado por cores: a cor vermelha significa que ocorreu erro durante a execução do serviço, a cor verde significa que o serviço executou com sucesso, a cor laranja mostra que o serviço ainda está executando e a cor azul informa que o serviço ainda não executou.

Os gráficos da Figura 9 representam serviços distintos, ou seja, cada valor apresentado é em relação a somente aquele serviço. O gráfico do SPService como está na cor verde e com o índice de 100% mostra ao usuário que o serviço foi executado com sucesso sem nenhum erro, já o serviço do RGM é apresentado ao usuário com erro e ainda sem finalizar.

O serviço de RGM são índices que o serviço deverá coletar do banco de dados dos clientes alimentando o banco de dados do franqueador. Essa rotina é executada diariamente e alguns dos índices coletados como quantidade de alunos novos, contratos vigentes, total de turmas servem como base para o franqueador avaliar como está o andamento dos seus franqueados.

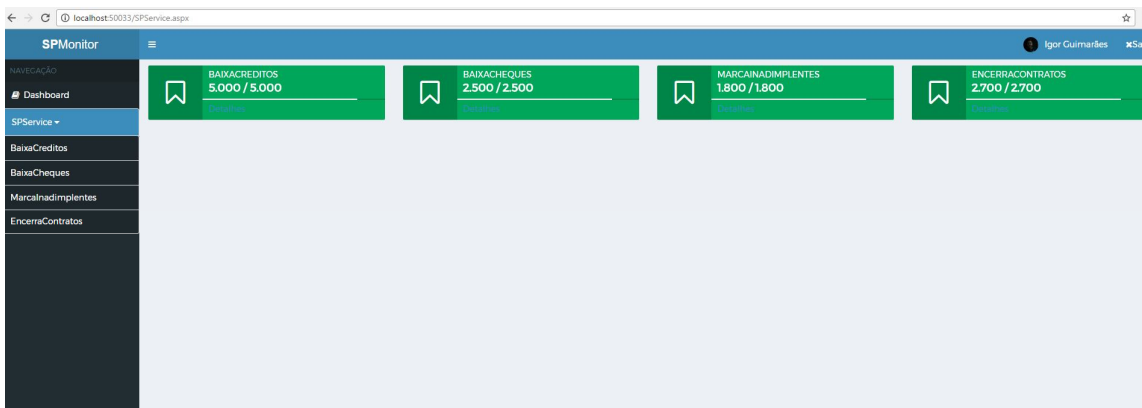
O SPService é rotina que os clientes optam se querem ou não que sejam executadas. Por exemplo o cliente pode optar por encerrar automaticamente contratos

de alunos quando chegar a data de término. Para isso ele deve habilitar uma opção que automaticamente vai habilitar esse serviço para ele, as outras rotinas dele funcionam dessa mesma forma.



**Figura 9 – Tela principal do sistema**

A Figura 10 apresenta a tela de detalhes do SPService após ser clicado no *hyperlink* (detalhes) o sistema redireciona o usuário para a página SPService.aspx. Foi criado um gráfico separado para rotina do SPService e todos os dados são coletados de suas respectivas tabelas do banco de dados.



**Figura 10 – Rotinas do SPService**

Na Figura 11 é apresentada a tela de detalhes de um serviço com seus respectivos índices. Cada índice informado foi definido previamente pelo DBA, o usuário (DBA) poderá realizar alguns filtros nessa tela, tais como por período ou código de cliente.

Os índices apresentados se referem às rotinas individuais que o SPService executa. O índice apresentado na baixa de créditos significa que foram baixados 5

mil créditos dos 5 mil que estavam previstos pelo serviço. Os índices das outras rotinas funcionam da mesma maneira.

Nome do Banco de Dados	Codigo de Cliente	Data de Inicio	Data de Término	Observação
SponteWeb1	101010	28/03/2017 15:46:22	28/03/2017 15:47:00	
SponteWeb2	2362	28/03/2017 15:46:22	28/03/2017 15:49:22	
SponteWeb3	257	28/03/2017 15:50:22	28/03/2017 15:51:22	
SponteWeb4	5287	28/03/2017 15:52:22	28/03/2017 15:53:22	
SponteWeb8	46355	28/03/2017 15:54:22	28/03/2017 15:55:22	
SponteWeb9	46860	28/03/2017 15:56:22	28/03/2017 15:57:22	
SponteWeb10	46252	28/03/2017 15:58:22	28/03/2017 15:59:22	
SponteWeb973	46964	28/03/2017 16:00:22	28/03/2017 16:01:22	
SponteWeb14	6043	28/03/2017 16:02:22	28/03/2017 16:03:22	
SponteWeb14	46179	28/03/2017 16:04:22	28/03/2017 16:05:22	

Figura 11 – Tela de detalhes do serviço

A coluna de observação do *grid* de detalhes do serviço somente será preenchida em caso de erro na execução do serviço. A mensagem de erro que será apresentada é a que consta na validação do código do sistema. Essas mensagens são, por exemplo, erros de desenvolvimento e informações de indisponibilidade de banco de dados.

#### 4.4 IMPLEMENTAÇÃO DO SISTEMA

O início do desenvolvimento ocorreu pela implementação da página mestre ou principal do sistema, mais conhecida como *master page*. Todas as outras páginas (exceto a página de *login.aspx*) herdam o que está na página principal. Em seguida foi desenvolvida a página de *login*. Para o usuário acessar o sistema, ele deverá ter seu usuário e senha previamente cadastrados. A Listagem 1 descreve a estrutura visual (HTML) da página de login do sistema.

```
<body style="padding-top: 20px;">
  <div class="container">
    <div class="row">
```

```

<div class="col-md-4 col-md-offset-4">
  <center style="padding-bottom: 20px">
    </center>
  <div class="panel panel-default">
    <div class="panel-heading">
      <h3 class="panel-title">
        Monitor de Recursos</h3>
      </div>
    <div class="panel-body">
      <form id="Form2" runat="server" accept-
charset="UTF-8" role="form">
        <fieldset>
          <div class="form-group">
            <asp:TextBox class="form-control"
runat="server" ID="txtUsuario" required Placeholder="Usuário" />
          </div>
          <div class="form-group">
            <asp:TextBox class="form-control"
runat="server" ID="txtSenha" required Placeholder="Senha"
            TextMode="Password" />
          </div>
          <asp:Button class="btn btn-lg btn-primary btn-
block" runat="server" ID="btnLogar"
            Text="Acessar" />
          <div style="padding-top: 10px;">
          </div>
          <div id="mensagem" />
        </fieldset>
      </form>
    </div>
  </div>
</div>
</div>
</body>

```

**Listagem 1 – HTML página de login**

Na Listagem 1, existe várias *divs* (divisões), classes (leiaute) e *forms* (formulário). Nessa listagem é feita a declaração do botão logar do sistema. No Visual Studio todos os componentes *buttons* possuem um evento chamado *OnClick*. Nesse evento é incluída toda a programação que é executada quando o usuário clicar nesse botão, o código dessa função está na Listagem 2.

```

Private Sub btnLogar_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles btnLogar.Click
    FormsAuthentication.SignOut()

```



```

        Dim dtUsuarios As DataTable = SPMonitor.RetDataTable("SELECT NOMEUSUARIO FROM
        USUARIOS WHERE NOMEUSUARIO = '" & txtUsuario.Text & "' AND SENHA = '" &
        txtSenha.Text & "'")
        If dtUsuarios.Rows.Count > 0 Then
            FormsAuthentication.RedirectFromLoginPage(txtUsuario.Text.ToUpper, True)
            Response.Redirect("Dashboard.aspx")
        Else
            MostraMensagem("Senha/Usuário inválidos. Tente novamente.", False)
        End If

    End Sub

    Private Sub MostraMensagem(ByVal sMsg As String, ByVal lSucesso As Boolean)
        Dim sTipo As String = "Sucesso!"
        Dim sClasse As String = "alert alert-success"

        If Not lSucesso Then
            sTipo = "Oops!"
            sClasse = "alert alert-danger"
        End If

        ScriptManager.RegisterStartupScript(Me.Page, Me.GetType, "addCls",
        "$('#mensagem').addClass('" & sClasse & "');" , True)
        ScriptManager.RegisterStartupScript(Me.Page, Me.GetType, "msg",
        "$('#mensagem').html('<a href=""#" class=""close"" data-
        di smi ss=""al ert""&times;</a><strong>' & sTipo & "</strong>' & sMsg & "');" , True)
    End Sub

```

#### Listagem 2 – Código da função de login

Na Listagem 2 é apresentando o código que faz a validação do *login*. Esse código é chamado sempre que o usuário clicar no botão acessar da página de *login*. É criado um *datatable* conforme mostra a linha (Dim dtUsuarios as DataTable) que verificará e retornará a quantidade de registros que existe na tabela de usuários com o usuário e senha informado na página de *login*.

Nesse código, também, é feita a validação para verificar a quantidade de registros obtidos pelos dados fornecidos (usuário e senha). Se o retorno do *datatable* for superior a 0, significa que existe usuário e senha conforme os dados informados, ou seja, o sistema permite que o usuário siga em frente e acesse a página principal (*dashboad.aspx*) conforme mostra o trecho *response.redirect*. Se a quantidade for 0 significa que não existe nenhum usuário e senha cadastrados no banco de dados e o sistema apresentará, por meio de uma função (conforme o trecho *private sub* mostramensagem), uma mensagem de senha e/ou usuário inválidos, conforme mostra a Figura 12.

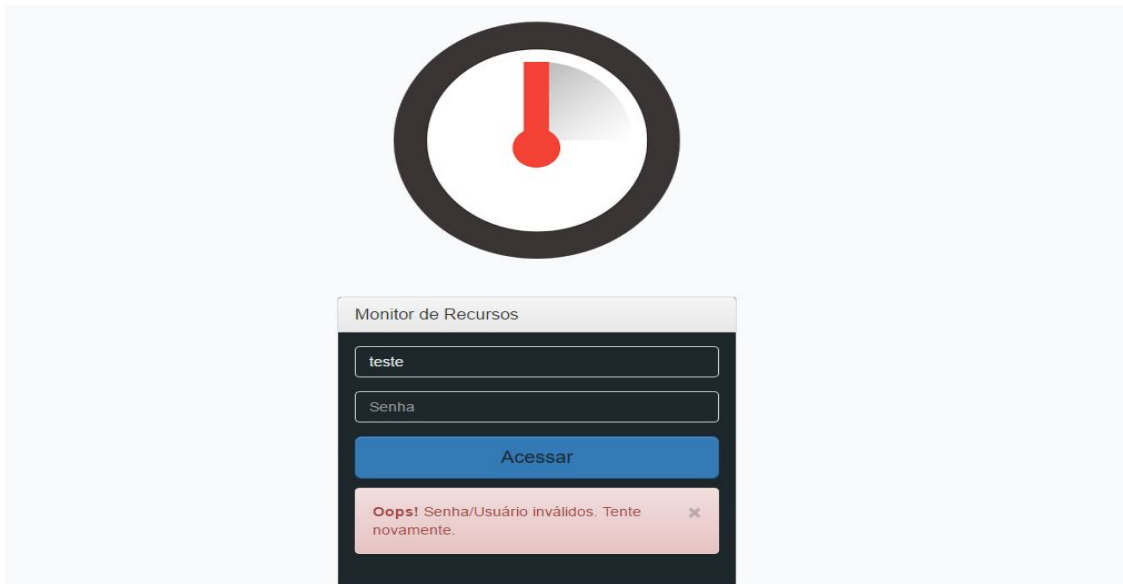


Figura 12 – Tela de validação do login (usuário e senha)

A Listagem 3 apresenta o código *JavaScript* responsável pela abertura da tela de cadastros em *modal*. Desta forma, permite-se a abertura de telas internamente na página que a chamou (*Master Page*), utilizando o componente *HighSlide*. Ao clicar no botão de inclusão ou edição as funções apresentadas na Listagem 3 são instanciadas para abrir a nova tela de cadastro em uma janela *modal*.

```

jQuery(document).ready(function () {
    jQuery('a[data-highslide="true"]').click(function () {
        var elementType = jQuery(this).get(0).tagName;
        var enabled = true;
        if (elementType == 'A') {
            enabled = (jQuery(this).attr('data-disabled') == 'true' ? false : true);
        }
        else {
            enabled = jQuery(this).prop('disabled');
        }
        if (enabled) {
            if (jQuery(this).attr('data-dependence') == 'grid')
            {
                $item = jQuery('table.table tbody
tr').not(":first").find('input:checked').first();
                if ($item.attr('accesskey') == undefined) {
                    alert('Selecione um item.');
```

```

        width: jQuery(this).attr('data-width'),
        height: jQuery(this).attr('data-height'),
        creditsPosition: 'bottom left',
        headingText: jQuery(this).attr('title'),
        wrapperClassName: 'titlebar'
    });
} else {
    return false;
}
});
});

```

**Listagem 3 – Método visualização detalhada dos serviços**

A Listagem 4 exibe alguns dos códigos CSS do Bootstrap utilizados para estilização dos componentes no sistema. Nele são definidas propriedades como tamanho, borda, cores, espaçamento e tamanho de fonte.

```

html {
    font-family: sans-serif;
    -webkit-text-size-adjust: 100%;
    -ms-text-size-adjust: 100%;
}
body {
    margin: 0;
}
article,
aside,
details,
figcaption,
figure,
footer,
header,
hgroup,
main,
menu,
nav,
section,
summary {
    display: block;
}
audio,
canvas,
progress,
video {
    display: inline-block;
    vertical-align: baseline;
}
audio:not([controls]) {
    display: none;
    height: 0;
}

```

**Listagem 4 – CSS do Bootstrap**

A Listagem 5 apresenta a função para coletar os índices do gráfico do RGM, para cada índice coletado foi necessário criar funções individuais coletando informações do banco de dados do sistema.

```

PROTECTED Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Me. Load
    VerificaStatus()
    VerificaTotalClientesColetas()
    VerificaTotalRestante()
    VerificaPorcentagem()
End Sub
Private Sub VerificaStatus()
    Dim Dt As DataTable = SPMonitor.RetDataTable("SELECT TOP(1) CLIENTES.NomeBD,
CLIENTES.CodigoCliente, ColetaRGM.DataInicio, ColetaRGM.DataTermino, ColetaRGM.Status
" & "FROM ColetaRGM WITH(NOLOCK) " & "INNER JOIN CLIENTES WITH(NOLOCK) On
CLIENTES.ClienteID = ColetaRGM.ClienteID " & "WHERE ColetaRGM.STATUS in (1,2) AND " &
"DAY(DataInicio) = day(getdate()) And Month(DataInicio) = Month(getdate()) And
YEAR(DataInicio) = year(getdate())", Nothing)
    If Dt.Rows.Count = 0 Then
        Exit Sub
    Else
        If Dt.Rows(0)("Status") = 1 Then
            divRgm.Attributes.Add("class", "small-box bg-orange")
        ElseIf Dt.Rows(0)("Status") = 2 Then
            divRgm.Attributes.Add("class", "small-box bg-red")
        Else
            divRgm.Attributes.Add("class", "small-box bg-green")
        End If
    End If
End Sub
Private Sub VerificaTotalClientesColetas()

    Dim TotalFranqueados As String = ""
    Dim Dt As DataTable = Franqueador.RetDataTable("SELECT COUNT (*) AS
TotalFranqueados FROM Franqueados WITH(NOLOCK) where Si tuacaoFranqueadoID = 1 and
codclisponde > 0 ", Nothing)

    TotalFranqueados = Dt.Rows(0)("TotalFranqueados")

    LblTotalFranqueados.Text = "Total de clientes a serem coletados: " &
TotalFranqueados
End Sub

Private Sub VerificaTotalRestante()

    Dim TotalRestante As String = ""
    Dim DtTotalFranqueador As DataTable = Franqueador.RetDataTable("SELECT COUNT
(*) AS TotalFranqueados FROM Franqueados WITH(NOLOCK) where Si tuacaoFranqueadoID = 1
and codclisponde > 0 ", Nothing)
    Dim DtTotalColetado As DataTable = SPMonitor.RetDataTable("SELECT COUNT(*) As
TotalColetado FROM ColetaRGM WITH(NOLOCK) " &
"WHERE
DAY(DataInicio) = day(getdate()) And Month(DataInicio) = Month(getdate()) And
YEAR(DataInicio) = year(getdate()) ", Nothing)

    TotalRestante = DtTotalFranqueador.Rows(0)("TotalFranqueados") -
DtTotalColetado.Rows(0)("TotalColetado")

```

```

End Sub

Private Sub VerificaPorcentagem()

    Dim TotalProcentagem As String = ""

    Dim DtTotal Franqueador As DataTable = Franqueador.RetDataTable("SELECT COUNT
(*) AS Total Franqueados FROM Franqueados WITH(NOLOCK) where Si tuacaoFranqueadoID = 1
and codclisponte > 0 ", Nothing)
    Dim DtTotal Coletado As DataTable = SPMonitor.RetDataTable("SELECT COUNT(*) AS
Total Coletado FROM ColetaRGM WITH(NOLOCK) " &
"WHERE
DAY(DataInicio) = day(getdate()) And Month(DataInicio) = Month(getdate()) And
YEAR(DataInicio) = year(getdate()) ", Nothing)
    TotalProcentagem = DtTotal Coletado.Rows(0)("Total Coletado") * 100 \
DtTotal Franqueador.Rows(0)("Total Franqueados")
    lblPocentagem.Text = TotalProcentagem

End Sub
End Class

```

#### Listagem 5 – Funções coletoras de índice

Conforme descreve a Listagem 5, existem quatro funções:

a) Verificastatus() - função responsável por identificar no banco de dados qual o *status* da coleta. Se o status for igual a 0, o gráfico receberá a cor verde, ou seja, a coleta finalizou sem nenhum problema; se o status for igual a 1, o gráfico receberá a cor laranja, significando que a rotina ainda está realizando as coletas; se o status for igual 2 o gráfico receberá a cor vermelha, significa que aconteceu algum problema na rotina de uma ou mais coletas.

b) VerificaTotalClientesColetados() - função responsável por identificar a quantidade de clientes que já foram coletados os índices e mostrar o total no gráfico do RGM.

c) VerificaTotalRestante() - função responsável por calcular a quantidade total de clientes que devem ser coletados subtraindo a quantidade de clientes já coletados e fornecer resultado no gráfico do RGM.

d) VerificaPorcentagem() - função responsável por calcular a porcentagem de coletas já realizadas e fornecer o índice no gráfico.

## 5 CONCLUSÃO

Tendo em vista a necessidade de avaliar as informações para instrumentar o DBA no processo de tomada de decisão, percebeu-se a necessidade do desenvolvimento de um software com as funcionalidades como proposto neste trabalho. Para que o aplicativo fosse desenvolvido visando atender aos objetivos definidos foi, inicialmente, necessário compreender todo o funcionamento do negócio para que as funcionalidades do sistema atendessem as regras de negócio envolvidas.

No desenvolvimento desse sistema em BI, poderão ser criados inúmeros serviços para coleta das informações. Cada um deles contendo informações que podem ou não ser iguais as dos outros. Muitos serviços que serão monitorados têm as suas particularidades e também será possível armazenar dados em grande escala.

O sistema apresentado já está sendo utilizado pela empresa para a qual ele foi desenvolvido. No momento o sistema possui as funcionalidades de monitoramento requeridas pela empresa. E todos os objetivos e requisitos mencionados nesse trabalho foram alcançados.

O sistema não foi desenvolvido em *Model-View-Controller* (MVC), método altamente recomendado atualmente no mercado, devido ao pouco tempo para conclusão do projeto e pela falta de experiência do desenvolvedor com o uso do MVC. Por trabalhar com *webforms* há mais de 5 anos foi optado por esse desenvolvimento.

Como perspectivas futuras para complementar as funcionalidades desenvolvidas para o software, pretende-se implementar rotinas para monitoramento de todos os mais de 30 serviços que estão em execução atualmente na empresa.

## REFERÊNCIAS

AAKER, David A.; KUMAR, Santhosh; DAY, George; LEONE, Robert. **Marketing research**, 10 ed. Wiley.com, 2009.

ALVAREZ, Miguel Angel. **O que é Javascript**. Disponível em <<http://www.criarweb.com/artigos/184.php>>. Acesso em: 12 mar. 2017.

AMARAL, Luís; VARAJÃO, João. **Planeamento de sistemas de informação**. 4 ed, FCA – Editora de Informática, 2007.

ASP NET, **Get started with ASP NET**. Disponível em:<<https://www.asp.net/get-started>>. Acesso em: 25 fev. 2017.

AVRAM, Abel. **Bootstrap 3: nova aparência e mais componentes**. Disponível em: <<http://www.infoq.com/br/news/2013/08/bootstrap-3>>. Acesso em: 12 mar. 2017.

BALSAMIQ, **Balsamiq Mockups, criando protótipos de telas para seus softwares**. Disponível em: < <http://blog.pierrefreire.com.br/2010/01/balsamiq-mockups-criando-prototipos-de.html>>. Acesso em: 20 mar. 2017.

BARBIERI, Carlos. **Business intelligence: modelagem & tecnologia**. Axcel Books, 2001.

BIZAGI. **Modelagem de Processos com Bizagi Modeler**. Disponível em: <<http://www.sgc.goias.gov.br/upload/arquivos/2014-10/manual-de-padronizacao-de-modelagem-de-processos-usando-bizagi---v3-1.pdf>>. Acesso em: 02 mar. 2017.

BOSE, Ranjit. Advanced analytics: opportunities and challenge. **Emerald Publishing: Industrial Management & Data Systems**, v.109, n.2, 2009.

CASE STUDIO, **Ferramenta de projeto de banco de dados**. Disponível em: <<http://www.casestudio.com/enu/index.aspx>>. Acesso em: 25 fev. 2017.

FLANAGAN, David. **Javascript o guia definitivo**. Porto Alegre: Editora ARTMED, 2002.

GOMES, Ana Laura. **XHTML/CSS criação de páginas web**. São Paulo: Editora Senac São Paulo, 2010.

GUARDA, Teresa; PINTO, Filipe Mota; CORDOVA, Juan Pablo; MATO, Fernando; Quiña, Geovanni Ninahualpa; AUGUSTO, Maria Fernanda. **Pervasive business intelligence as a competitive advantage**. In: 11th Iberian Conference on Information Systems and Technologies (CISTI), 2016, p. 1-4.

HMTL 5. **HTML5 Introduction**. Disponível em:  
<[https://www.w3schools.com/html/html5\\_intro.asp](https://www.w3schools.com/html/html5_intro.asp)>. Acesso em: 21 mar. 2017.

JAVASCRIPT. **JavaScript**. Disponível em: <<https://www.javascript.com/>>. Acesso em: 21 mar. 2017.

JQUERY. **Jquery**. Disponível em: <<https://jquery.com/>>. Acesso em: 21 mar. 2017.

LI, Kin Fun; CAMPBELL, Graham; KUMKA, Allan; NESMITH, Chris. **The Design and Implementation of a Business Intelligence Recommender**. In: 2011 International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 2011, p. 219-224.

MANAGEMENT STUDIO, **SQL Server Management Studio**. Disponível em:  
<<https://msdn.microsoft.com/pt-BR/library/mt238290.aspx>>. Acesso em: 02 mar. 2017.

MARINHEIRO, António; BERNARDINO, Jorge. **Analysis of open source business intelligence suites**, In: 8th Iberian Conference on Information Systems and Technologies (CISTI), 2013, p. 1-7.

MARQUES, Henrique. **Twitter Bootstrap: Aumentando sua produtividade**. Devmedia. Disponível em: <<http://www.devmedia.com.br/seja-produtivo-usando-o-twitter-bootstrap/24967#ixzz2ecT0pagb>>. Acesso em: 21 mar. 2017.

MILLER, Stephen; MCROBBIE, Graeme; BEGG, Carolyn. **Business intelligence tools: should they be 'gamified'?** Project 'GamBIT': Evaluating User Engagement. In: SAI Computing Conference 2016, p. 758-763.

MOREIRA, Daniela. **Aplicações: AJAX em ação**. IDGNOW. Disponível em:<<http://idgnow.com.br/internet/2005/10/19/idgnoticia.2006-03-12.8450093052/#&panel2-2&panel1-1>>. Acesso em: 05 mar. 2017.

MSDN MICROSOFT. **Visão Geral do Servidor Web (IIS)**. Microsoft. 2016. Disponível em: <[https://msdn.microsoft.com/pt-br/library/hh831725\(v=ws.11\).aspx](https://msdn.microsoft.com/pt-br/library/hh831725(v=ws.11).aspx)>. Acesso em: 01 mar. 2017.

MSDN MICROSOFT. **Visão Geral do .NET Framework**. 2017. Disponível em: <<http://msdn.microsoft.com/pt-br/library/vstudio/zw4w595w.aspx>>. Acesso em: 04 mar. 2017.

NET BEANS, **Introdução ao Ajax para Aplicações Java Web**, Disponível em: <[https://netbeans.org/kb/docs/web/ajax-quickstart\\_pt\\_BR.html](https://netbeans.org/kb/docs/web/ajax-quickstart_pt_BR.html)>. Acesso em: 04 mar. 2017.



ORACLE DATABASE 11g for Data Warehousing and Business Intelligence. **Oracle White Paper**, 2009. Disponível em:  
<<http://www.oracle.com/technetwork/database/bi-datawarehousing/twp-bidw-overview-11gr1-133677.pdf>>. Acesso em: 18 set. 2016.

PAULINO, Daniel. **Saiba o que é um Windows Service ou simplesmente Serviço do Windows**. 2009. Disponível em:  
<[https://www.oficinadanet.com.br/artigo/2061/saiba\\_o\\_que\\_e\\_um\\_windows\\_service\\_ou\\_simplesmente\\_servico\\_do\\_windows](https://www.oficinadanet.com.br/artigo/2061/saiba_o_que_e_um_windows_service_ou_simplesmente_servico_do_windows)>. Acesso em: 18 set. 2016.

POWER, Daniel J. **A brief history of decision support systems**. 4 ed. 2007. DSSResources.COM. World WideWeb, Disponível em:  
<<http://dssresources.com/history/dsshhistory.html>>. Acesso em: 27 set. 2016.

SENIOR. **Business intelligence**. e-book. Disponível em:  
<<http://www.magnus.com.br/display/4c6ffa1c-6e9d-4af0-8a6a-f9af38d15cea>>. Acesso em: 21 set. 2016.

SEZÕES, Carlos; OLIVEIRA, José; Baptista, Miguel. **Business Intelligence**. SPI – Sociedade Portuguesa de Inovação, 2006. Disponível em:  
<[http://www.bi4all.pt/docs/Manual\\_Business\\_Intelligence.pdf](http://www.bi4all.pt/docs/Manual_Business_Intelligence.pdf)>. Acesso em: 20 fev. 2017.

SILVA, Samy. **Site sobre CSS e padrões web**. Disponível em:  
<<http://www.maujor.com/index.php>>. Acesso em: 9 out. 2016.

SQL SERVER, **Microsoft® SQL Server® 2014 Express**. Disponível em:  
<<https://www.microsoft.com/pt-br/download/details.aspx?id=42299>>. Acesso em: 25 mar. 2017.

STACKOWIAK, Robert; RAYMAN, Joseph; GREENWALD, Rick. **Oracle data warehousing and business intelligence solutions**. Wiley Publishing: Indianapolis, 2007.

TURBAN, Efrain; SHARDA, Ramesh; DELEN, Dursun; KING, David. **Business intelligence: a managerial approach**. 2 ed. New Jersey, Prentice Hall, 2011.

VB NET, **Introdução ao Visual Basic.NET**. Disponível em:  
<<http://msdn.microsoft.com/pt-br/library/cc580665.aspx>>. Acesso em: 25 fev. 2017.

VISUAL STUDIO, **Ferramentas para todos os desenvolvedores e aplicativos**. Disponível: <<https://www.visualstudio.com/>>. Acesso em: 23 set. 2016.

XIE, Guotong; YANG, Yang Y.; LIU, Shnegping; QIU, Zhaoming; PAN, Yue; XIONGZHI, Zhou. EIAW: towards a business-friendly data warehouse using semantic web technologies. **The Semantic Web**, p. 857-870, 2007.

YAMAZACK, Wesley. **Quick Tips: Case Studio 2 - Parte I**. Devmedia. Disponível em: <<http://www.devmedia.com.br/quick-tips-case-studio-2-parte-i/16100>>. Acesso em: 25 fev. 2017.