

APLICAÇÃO WEB PARA MONITORAMENTO DE VENDAS NO VAREJO

João Rafael Colombo

Faculdades Integradas de Taquara – Faccat – Taquara – RS – Brasil
joacolombo@aluno.faccat.br

Francisco Assis Moreira do Nascimento

Professor Orientador

Faculdades Integradas de Taquara – Faccat – Taquara – RS – Brasil
assis@faccat.br

Resumo

Este trabalho apresenta uma aplicação Web, que permite medir o desempenho de uma loja de varejo, utilizando a gestão de indicadores específicos para este tipo de loja, tais como, preço médio por peça vendida (PMPV), número de peças por venda (PV) e ticket médio. A aplicação Web coleta os dados diretamente da base de dados do sistema de gestão da empresa, organizando-os em uma estrutura, em que é possível acompanhar o desempenho de produtos, grupo de produtos, subgrupo de produtos, loja e rede de lojas, bem como, o desempenho dos vendedores. O sistema se destaca pela facilidade de uso, fazendo com que as informações sobre o desempenho fiquem bem estruturadas e de fácil entendimento. Um protótipo da aplicação Web foi desenvolvido em C#, em ambiente Visual Studio 2015, com servidor IIS 8 hospedado na Amazon, utilizando para apresentação o ASP.NET Razor e para persistência o *Entity Framework* com SQL Server Express 2014, se comunicando com o banco de dados do sistema de gestão Linx. O protótipo tem sido utilizado em uma empresa calçadista da região do Vale do Paranhana, com bons resultados reportados pelos gestores das lojas.

Palavras-chave: *aplicação Web, varejo, monitoramento de vendas.*

WEB APPLICATION FOR RETAIL SALES MONITORING

Abstract

This paper presents a web application oriented to the performance evaluation of retail stores by using the management of specific indicators, such as, average price per sold piece, number of pieces per sale, and average ticket. The Web application collects data directly from the enterprise resource planning system's database, adopting a structure that makes it possible to monitor easily the performance of products, group of products, subgroup of products, store and retail chain, as well as, the performance of sellers. The web application is very easy to operate, by showing all the information in a very organized and structured manner. A prototype of the Web application was implemented in C#, Visual Studio 2015, under an ISS 8 web server, hosted on Amazon, using ASP.NET Razor for the view layer and Entity Framework with SQL Server Express 2014 for persistence layer, which communicates with the database server of a Linx Enterprise Resource Planning. The Web application prototype has been successfully used in a footwear company at the Vale do Paranhana region, with good results being reported by the retail managers..

Key-words: *Web application, retail, sales management.*

1. INTRODUÇÃO

Atualmente, os gestores de lojas de varejo enfrentam muitas dificuldades para conseguir tomar decisões acertadas para o negócio, pois precisam lidar com uma imensa quantidade de informações, referentes aos produtos, vendedores, vendas, preços e condições. (MATTAR, 2011).

Assim, sem ferramentas eficientes que ajudem os gestores, através da automação das tarefas de gestão e do tratamento adequado das informações, torna-se praticamente impossível superar estas dificuldades.

Um dos grandes problemas do varejo brasileiro, notadamente nas pequenas e médias operações, é a falta de indicadores de desempenho para que o empresário possa avaliar, periodicamente, o resultado de suas operações para estabelecer estratégia para melhoria.

Um dos grandes desafios da gestão de empresas é justamente se implantar um gerenciamento de indicadores de desempenho para avaliar, periodicamente, o resultado de suas operações e assim, estabelecer estratégias de melhoria. (CHAVES, 2008).

E um dos princípios da gestão de qualidade fala exatamente sobre isso: o que não se mede não se melhora (LORINO, 1992). Por esta razão, é de fundamental importância para uma operação varejista, a adoção de alguns indicadores de desempenho, escolhidos de acordo com as características da loja.

A aplicação Web proposta neste trabalho permite medir o desempenho de uma loja de varejo através da gestão de indicadores de desempenho. Isso será feito coletando os dados diretamente do banco de dados, organizando-os em uma estrutura onde será possível medir o desempenho de um produto específico, grupo de produtos, subgrupo de produtos, vendedores, loja e rede de lojas. O sistema primará pela facilidade de uso, fazendo com que a informação fique bem estruturada e de fácil entendimento.

Gestores de empresas no varejo necessitam de ferramentas cada vez mais personalizadas para monitorar as vendas, clientes, estoque e custos. Essas informações devem estar disponíveis de forma fácil e rápida para que o processo de tomada de decisão seja feito de forma acertada.

Decisões necessitam de um suporte informativo adequado, de maneira que a decisão seja tomada de maneira assertiva para a empresa. Para isso, os administradores necessitam de um sistema que lhes forneça informações sobre o desempenho realizado e metas esperadas, permitindo-lhes fazer comparações em bases objetivas. (RODRIGUEZ, 2004)

O uso de uma aplicação Web que ofereça estas funcionalidades para os gestores de empresas no varejo torna rápido e fácil o processo de tomada de decisões estratégicas que melhorem o desempenho das vendas da loja.

O presente trabalho trata justamente do desenvolvimento de uma aplicação Web, compatível com o sistema de venda de loja Linx, em que é possível medir o desempenho da loja e realizar uma gestão eficiente das vendas e facilitando a tomada. Além disso, a aplicação Web desenvolvida permite o cadastro de metas, para o lojista poder medir o desempenho da loja em relação a vendedores, produtos, grupo e subgrupo de produto, loja e rede de loja.

Na próxima Seção serão apresentados os conceitos e tecnologias empregados na realização do trabalho. Na Seção 2 é descrita a aplicação Web para monitoramento de desempenho no varejo, que foi implementada. Na Seção 3 são apresentados alguns usos do sistema, demonstrando a sua efetividade. A Seção 4 mostra os resultados obtidos com o uso do sistema e na última Seção são feitas algumas considerações finais como conclusão do trabalho.

1.1 Conceitos e Tecnologias

1.1.1 Indicadores de desempenho no varejo

Um dos princípios básicos da gestão de qualidade é “o que não se mede não se melhora”. Por esta razão é de fundamental importância, para uma operação varejista, a adoção de alguns indicadores de desempenho, escolhidos de acordo com as características da loja.

(LORINO, 1992)

Os indicadores de desempenho mais comuns, aplicados em qualquer operação varejista são os seguintes: (MATTAR, 2011)

- **Número de Peças por Venda (PV)** O objetivo deste indicador é verificar quantas peças foram vendidas por venda. A sua fórmula é Número de peças vendidas (P) no período dividido pelo número de Vendas (V) no mesmo período ($PV = P/V$)
- **Preço Médio por peça vendida (PMPV)** - Este indicador serve para indicar o preço médio das peças vendidas em um período. O seu cálculo é obtido pela fórmula: Valor total de venda (PM) por período dividido pelo número de peças vendidas (PV) no mesmo período ($PVPM = PV/PM$);

- **Preço médio por Venda (PMV) ou Ticket Médio**, tem como objetivo identificar o preço médio das vendas. O seu cálculo é obtido pela fórmula: Venda total (V) do período dividido pelo número de vendas (PM) efetuadas ($PMV=V/PM$).

Esses indicadores devem ser apurados, por vendedor, por loja, por área de supervisão ou de qualquer outra forma de acordo com as características da operação. O importante é que sejam acompanhados de forma sistemática e sendo melhor ainda quando se faz um acompanhamento diário, para que possa ser estabelecida uma estratégia para melhoria dos resultados. (TEIXEIRA, 2014)

Os indicadores são fundamentais não só para avaliar o desempenho, eles auxiliam a tomada de decisão de forma inteligente. Com isso, é possível apontar rapidamente quais indicadores estão puxando para baixo os resultados de uma loja, equipe, etc. Também é possível identificar possíveis problemas com a análise dos indicadores. (CHAVES, 2008)

1.1.2 Sistema de apoio à decisão

Um Sistema de Apoio à Decisão (SAD) é qualquer sistema que fornece ao usuário um apoio ao processo decisório. Normalmente, sistemas de apoio à decisão são sistemas complexos, que modelam os dados e permitem ao usuário analisar informações de um, ou mais sistemas, de forma unificada e simplificada para que se possa tomar a decisão mais assertiva. (PRIMAK, 2008)

Os principais objetivos dos sistemas de informação de apoio à tomada de decisões são monitorar os resultados de uma empresa e o seu ambiente externo, buscar e captar, avaliar e selecionar, tratar, condensar, indexar, analisar, interpretar, classificar, armazenar, recuperar, transmitir e disseminar dados e informações externas e internas à empresa, pertinentes e relevantes para a tomada de decisões relativas às diversas atividades. (MATTAR, 2011)

1.1.3 Principais Tecnologias Adotadas

Nesta Seção são apresentadas as principais metodologias, tecnologias e ferramentas que foram utilizadas na realização do presente trabalho.

Padrão MVC - Model, View, Controller

MVC (Model-View-Controller) é um padrão de arquitetura para o desenvolvimento de software, que separa as regras e lógicas do negócio, do fluxo da aplicação e da apresentação

permitindo um maior controle sobre a aplicação, possibilitando uma melhor manutenção do código e melhorando a segurança da aplicação. (SANCHEZ, 2013)

O principal objetivo do MVC é separar dados, ou lógica de negócios, (*Model*) da interface do usuário (*View*) e o fluxo da aplicação (*Controller*). O *Model* guarda os dados e informa suas *Views* e *Controllers* associados quando há uma mudança em seu estado, desta forma quando as *Views* recebem uma mudança de estado do *Model*, ela se atualiza deixando os seus dados corretos, e também permitindo que os controladores alterem o conjunto de comandos disponíveis. O *Controller* é quem faz a principal ligação entre a *View* e o *Model*, enviando informações para o *Model* alterar seu estado por exemplo, e para a *View* alterar a apresentação. A *View* gera uma visão das informações do *Model*. (SANCHEZ, 2013)

Metodologia DDD – Domain Driven Design

Domain Driven Design significa Projeto Orientado a Domínio. O nome vem do título do livro de Eric Evans (EVANS, 2005), empresário dono da *Domain Language*, uma empresa especializada em treinamento e consultoria para desenvolvimento de software. O livro de Evans apresenta vários padrões de projetos, baseados em experiências de Evans ao longo de mais de duas décadas de desenvolvendo software utilizando técnicas de Orientação a Objeto.

Algumas das principais características desta metodologia, que se deve destacar, incluem:

- Alinhamento do código com o negócio: o contato dos desenvolvedores com os especialistas do domínio é algo essencial quando se faz DDD;
- Favorecer reutilização: os blocos de construção, que serão vistos adiante, permitem aproveitar um mesmo conceito de domínio ou um mesmo código em vários lugares;
- Mínimo de acoplamento: Com um modelo bem feito, organizado, as várias partes de um sistema interagem sem que haja muita dependência entre módulos ou classes de objetos de conceitos distintos;
- Independência da Tecnologia: DDD não foca em tecnologia, mas sim em se entender as regras de negócio e como elas devem estar refletidas no código e no modelo de domínio. Não que a tecnologia usada não seja importante, mas essa não é uma preocupação de DDD.

Uma vez que se decide criar um modelo usando DDD, faz-se necessário, inicialmente, isolar o modelo de domínio das demais partes que compõem o sistema. Essa separação pode ser feita utilizando-se uma arquitetura em camadas, que dividirá a aplicação em quatro partes:

- Interface de Usuário – parte responsável pela exibição de informações do sistema ao usuário e também por interpretar comandos do usuário;
- Aplicação – essa camada não possui lógica de negócio. Ela é apenas uma camada fina, responsável por conectar a Interface de Usuário às camadas inferiores;
- Domínio – representa os conceitos, regras e lógicas de negócio. Todo o foco de DDD está nessa camada. Neste trabalho, uma das tarefas mais importantes será aperfeiçoar e compreender profundamente essa parte do sistema;
- Infraestrutura – fornece recursos técnicos que darão suporte às camadas superiores. São normalmente as partes de sistemas responsáveis por persistência de dados, conexões com bancos de dados, envio de mensagens por redes, gravação e leitura de discos, etc.

Linguagem C# e Framework .Net

Visual C# é uma implementação da linguagem C# pela Microsoft (ALBAHARI, 2010). Visual Studio suporta o Visual C# com um editor de código completo, compilador, modelos de projetos, designers, assistentes de código, um depurador poderoso e fácil de usar e outras ferramentas.

A biblioteca de classes do framework .Net fornece acesso a vários serviços do sistema operacional e outras classes úteis e bem estruturadas que aceleram significativamente o ciclo de desenvolvimento. (ALBAHARI, 2010)

Razor

O ASP.NET Razor é uma view engine que já está incluída no WebMatrix, com ele temos a possibilidade de inserir a lógica da aplicação diretamente na camada de visualização do projeto, por exemplo, é possível inserir a sintaxe razor junto com os códigos HTML dentro da mesma página, a sintaxe do ASP.NET razor é extremamente simplificada, baseado na linguagem C# .NET, embora seja possível utilizar a linguagem Visual Basic .NET. (SATO, 2015)

Visual Studio 2015 Community

A IDE Visual Studio é uma ferramenta interativa disponibilizada pela Microsoft, para o desenvolvimento de programas. A versão *Community* é gratuita e vem com flexibilidade para qualquer plataforma, ferramentas para melhorar a produtividade, milhares de extensões, e com muitas linguagens de programação como, C#, Visual Basic, F#, C++, HTML, e muitas outras. (MICROSOFT)

SQL Server 2014 Express

O SQL Server é uma ferramenta para armazenamento de dados disponibilizada pela Microsoft. A versão Express é gratuita podendo armazenar até 10GB de dados sem custo. Além de um editor de código SQL, a ferramenta é altamente escalável, tem alta disponibilidade, segurança e conformidade, entre outros recursos para facilitar o desenvolvimento e gerenciamento de programas (MICROSOFT).

1.2 Trabalhos Relacionados

Já existem no mercado alguns sistemas similares ao desenvolvido no presente trabalho, como o *Business Intelligence* para LS Retail da Empresa Targit e o Ascent Business Intelligence.

1.2.1 Sistema de BI do LS Retail

O *Business Intelligence* da empresa Targit foca no produto LS Retail, que é uma empresa terceira que apresenta uma solução para o varejo. (TARGIT, 2016) O sistema desenvolvido no presente trabalho se assemelha muito com o sistema da Targit, pois ambos são sistemas desenvolvidos para se acoplar em sistemas de terceiros, que fazem a gestão de varejo; porém, a principal diferença entre eles é a proporção: enquanto a Targit necessita de um “warehouse”¹ robusto, o sistema deste trabalho processa os dados on-line, e assim não necessita de uma infra-estrutura robusta para ser utilizado.

1.2.2 Sistema de BI da Ascent

O *Business Intelligence* da empresa Ascent (ASCENT, 2016) é uma ferramenta completa que se assemelha muito com o sistema desenvolvido no presente trabalho, pois ambos efetuam o processamento de dados on-line, não dependendo da infraestrutura do local;

porém, enquanto é possível tratar vários sistemas com o *Business Intelligence* da Ascent de forma genérica, o sistema deste trabalho é mais específico, focando somente na ferramenta Linx, tornando-o muito mais detalhado.

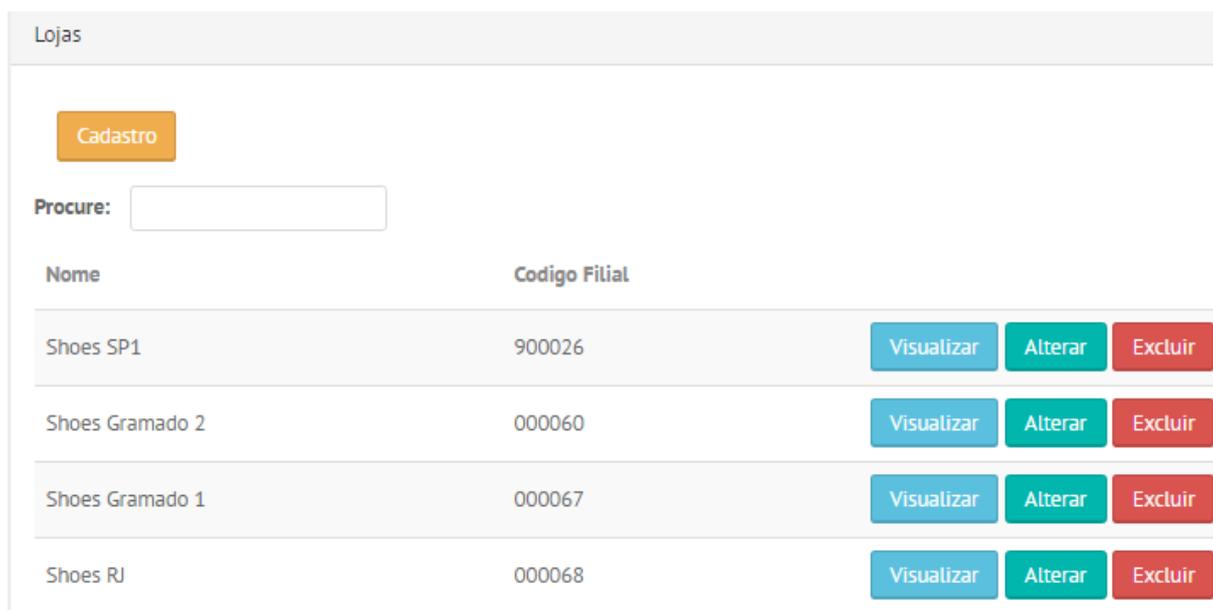
2 SISTEMA PARA MONITORAMENTO DE DESEMPENHO NO VAREJO

2.1 Visão Geral

As principais funcionalidades oferecidas pelo sistema para monitoramento de desempenho no varejo, desenvolvido neste trabalho, incluem:

- Manter loja: Após o usuário *logar*, é possível adicionar lojas, editar os atributos e excluir. A loja é cadastrada no sistema web antes do sincronismo de informações, pois existe uma separação de vendas por loja.(Figura 1)

Figura 1



Nome	Codigo Filial	Visualizar	Alterar	Excluir
Shoes SP1	900026	Visualizar	Alterar	Excluir
Shoes Gramado 2	000060	Visualizar	Alterar	Excluir
Shoes Gramado 1	000067	Visualizar	Alterar	Excluir
Shoes RJ	000068	Visualizar	Alterar	Excluir

Na Figura 1, está apresentada a listagem de lojas cadastradas, onde pode-se visualizar os dados da loja, alterar informações ou excluir a loja.

- Manter Rede: A gestão da rede é feita pelo mantedor do sistema. É entregue ao cliente a rede com os dados cadastrados e um usuário e senha para que ele possa cadastrar os outros usuários.
- Manter Lojista: Após o Usuário *logado*, é possível adicionar um novo usuário, editar e excluir outros usuários.
- Coletar dados do DB da loja: O sistema deve coletar os dados no banco de dados da loja (off-line), adicionar as informações referentes ao usuário, e inserir os dados coletados

no servidor online. Caso a loja perca conexão com a internet o sistema deve recolher os dados de forma retroativa para que não fiquem dados faltando no sistema on-line. Também é possível escolher uma data retroativa para o envio das informações.

- Medir desempenho de vendas: O sistema verifica quais lojas pertencem ao usuário, e coleta os dados de venda e meta do servidor online e apresentar para o usuário de forma gráfica e clara.
- Medir de carteira de clientes: O sistema verifica quais lojas pertencem ao usuário, e coleta os dados de clientes do servidor online e apresentar para o usuário de forma gráfica e clara.
- Manter Metas: No ambiente web, é ser possível determinar metas, para loja, para o vendedor, por grupo de produto, para ticket médio valor e quantidade de peças, quantidade de clientes cadastrados, estas metas podem ser exibidas no *dashboard* e possuem marcações claras indicando o estado da meta e se ela está correndo conforme o previsto.(Figura 2)

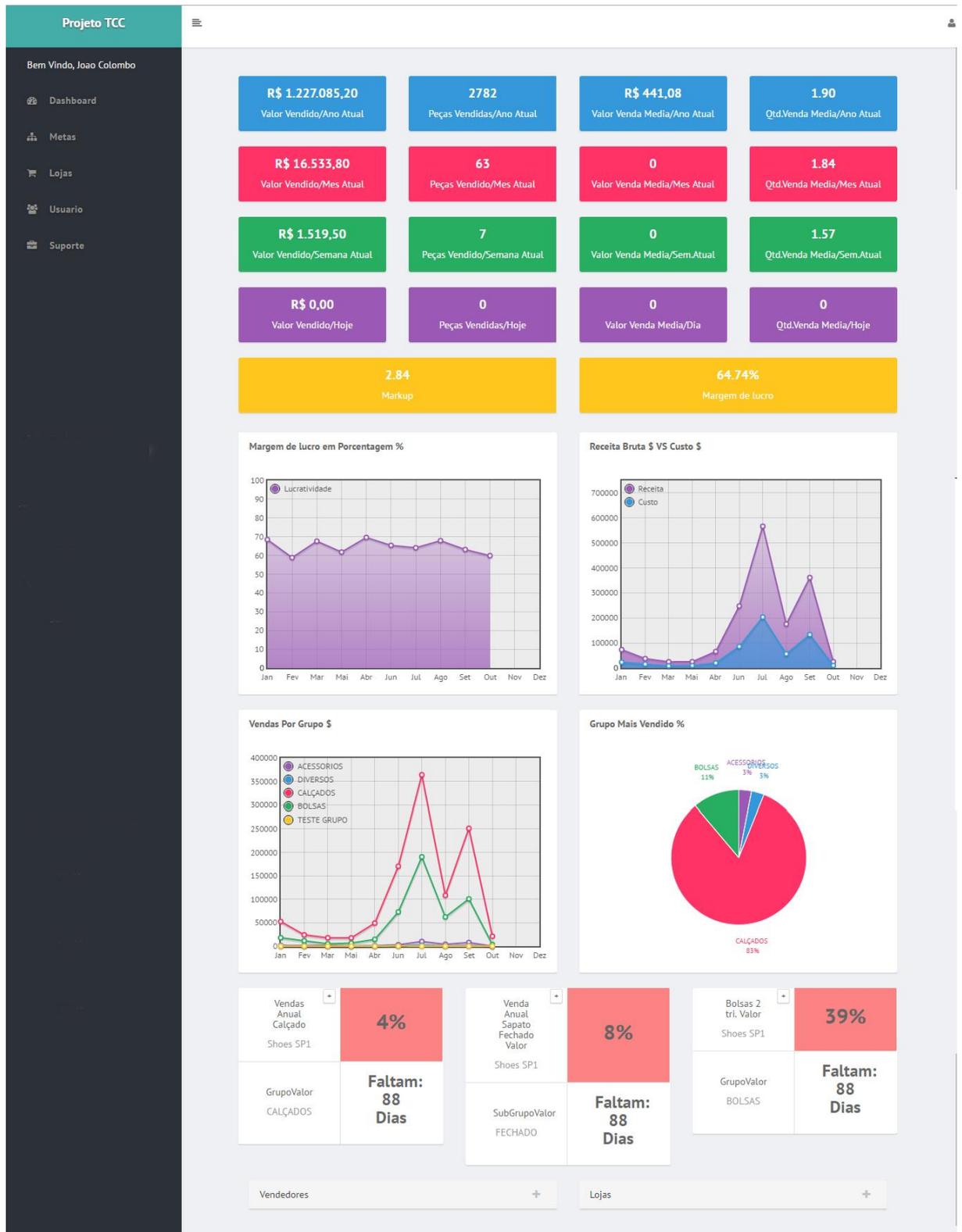
Figura 2

Nome	Progresso	%	Status	Data Final	Medida	Parametro	Loja	Visualizar	Excluir
Vendas Anual Calçado	<div style="width: 4%;"></div>	4%	Em Progresso	31/12/2016	GrupoValor	CALÇADOS	Shoes SP1	Visualizar	Excluir
Venda Mesal Julho Calçados	<div style="width: 3%;"></div>	3%	Encerrado	31/07/2016	GrupoValor	CALÇADOS	Shoes SP1	Visualizar	Excluir
Venda Anual Sapato Fechado Valor	<div style="width: 8%;"></div>	8%	Em Progresso	31/12/2016	SubGrupoValor	FECHADO	Shoes SP1	Visualizar	Excluir
Bolsas 2 tri. Valor	<div style="width: 39%;"></div>	39%	Em Progresso	31/12/2016	GrupoValor	BOLSAS	Shoes SP1	Visualizar	Excluir

A Figura 2 apresenta as metas de venda, onde tem-se o nome da meta, o quanto já foi percorrido, o status e algumas informações como nome da loja e o que está sendo medido. Existe também a possibilidade de visualizar mais detalhes e excluir a meta.

O *dashboard* é o principal recurso do sistema, nele a informação está centralizada dando uma visão geral do estado atual dos indicadores do negócio e as metas que devem ser acompanhadas.

Figura 3



Na Figura 3 é apresentado o *dashboard* do sistema, que é a tela principal, reunindo todas as principais informações para os gestores. Na parte superior, tem-se os indicadores anuais, mensais, semestrais e diários do andamento da rede. Logo abaixo tem-se o markup e

margem de lucro média da rede. São apresentados também alguns gráficos, como por exemplo, o gráfico de margem de lucro por porcentagem, que apresenta o andamento anual da margem de lucro da rede. Próximo ao final da tela, tem-se as metas, que no momento do cadastro foram marcadas para exibição no *dashboard*. Nelas tem-se alguns dados, tais como, a quantidade de dias que faltam para o término, e se a meta está dentro do previsto. Caso ela esteja dentro do previsto, o quadrado contendo o percentual fica verde, caso contrário ficará vermelho.

2.2 Desenvolvimento do Sistema

No desenvolvimento do sistema foi adotado o método ágil *Scrum* com 5 interações. Para gerenciar as tarefas e interações, foi utilizado o *Team Foundation Server* (TFS) da *Microsoft*, que já é integrado com a IDE utilizada na codificação, *Visual Studio 2015*.

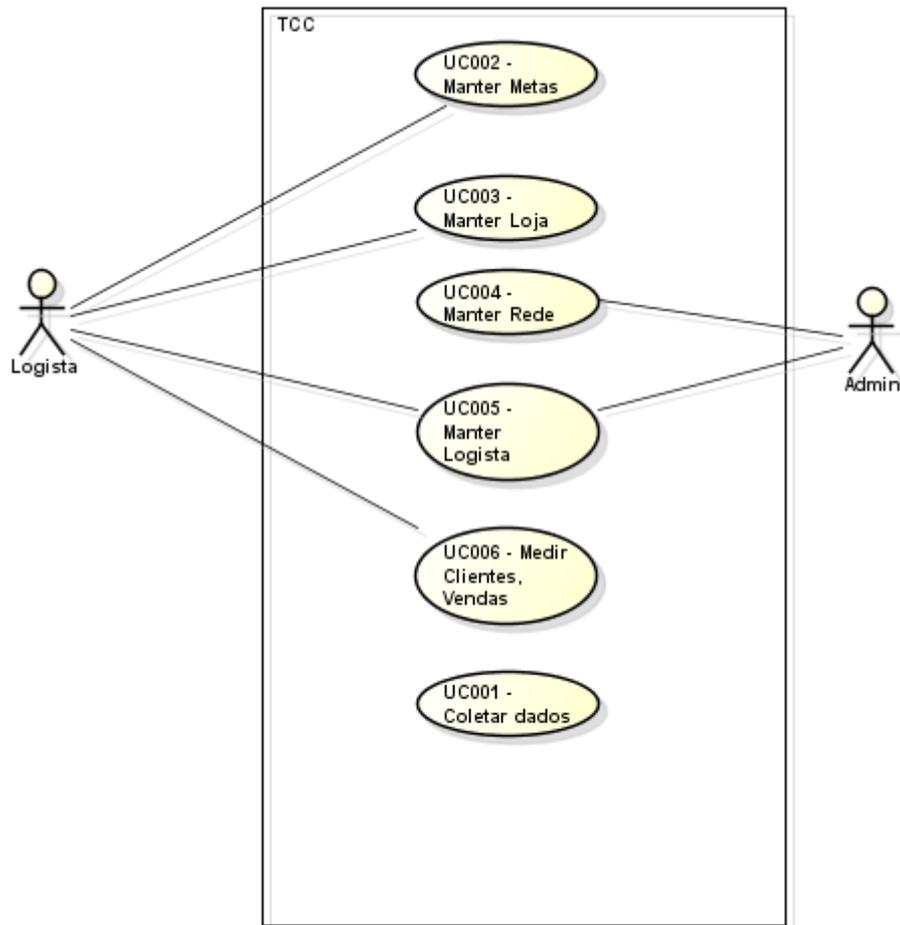
As interações foram organizadas conforme a complexidade e necessidade de informação. No primeiro Sprint, foi desenvolvido o sistema que é instalado na loja responsável por enviar a informação para o sistema on-line. No segundo Sprint, foi criada a parte de metas e uma parte do *dashboard*. No terceiro Sprint, foi desenvolvido o *dashboard*. No quarto Sprint, foi feita a separação de rede e cadastro lojas. No quinto Sprint, foi desenvolvido o cadastro de lojista e criada a parte de acesso ao sistema. Nesta etapa também foram realizados ajustes para melhorar a performance da ferramenta on-line.

2.2.1 Análise

A Etapa de análise começou com o levantamento de requisitos, verificando quais seriam as funcionalidades necessárias para atender ao problema. Isso foi realizado através de entrevistas com usuários e análise do processo que era adotado.

Após o levantamento de requisitos, mostrou-se necessária a criação de uma aplicação para atender à necessidade. A partir deste ponto foi feito o diagrama de caso de uso, como apresentado na Figura 4.

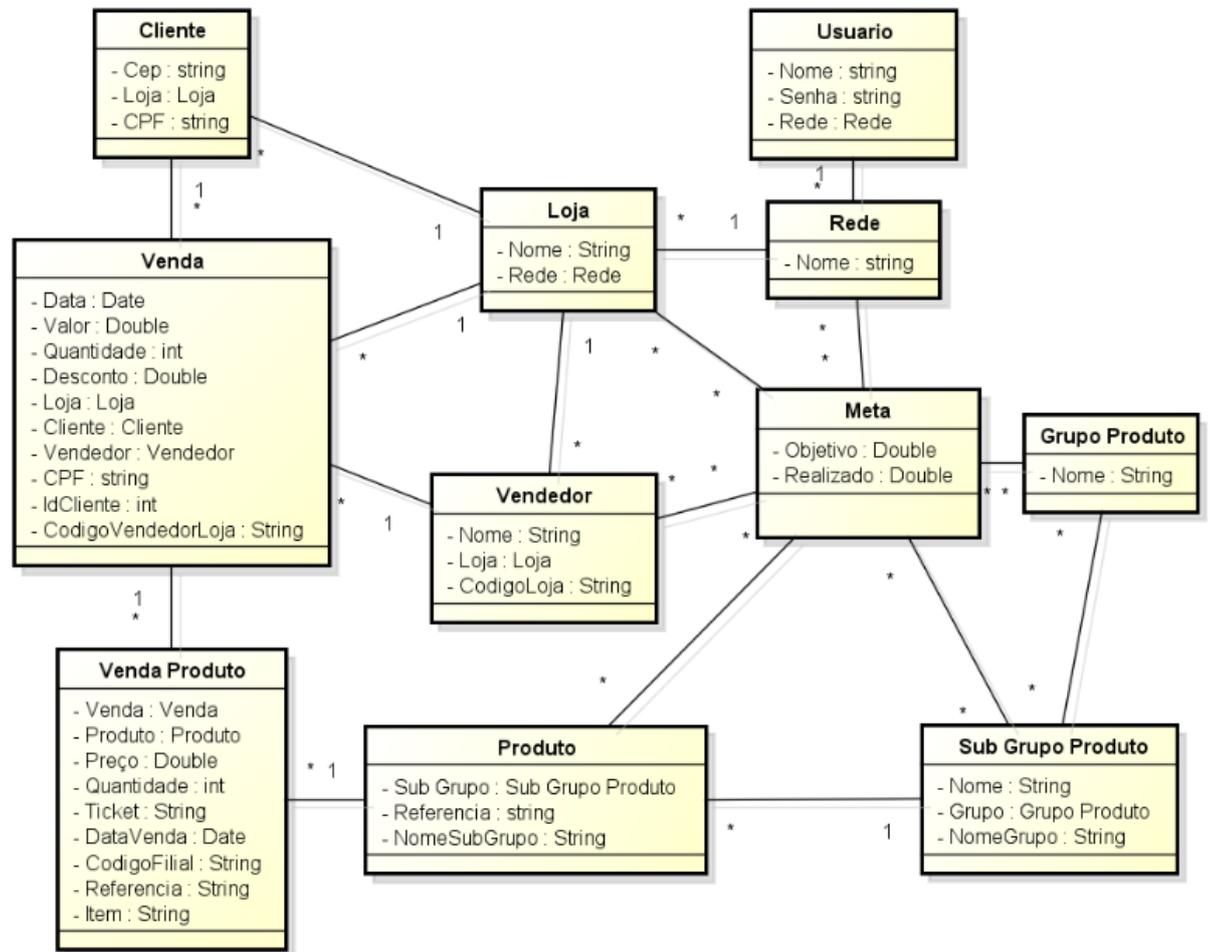
Figura 4



Com as informações coletadas no levantamento de requisitos, foi possível desenvolver o diagrama de caso de uso, mostrado na Figura 4. O usuário Lojista pode cadastrar lojas, onde é possível se medir o desempenho da mesma, cadastrando-se e monitorando-se metas e medidas. O lojista também pode cadastrar outros lojistas para acessar a sua rede de lojas. O Admin é o representante do sistema que faz o cadastro da rede e cadastra o primeiro lojista para que seja possível ao cliente acessar o sistema. A coleta de dados é feita automaticamente do computador da loja, não necessitando a interação com usuário pois ela ocorre de forma automática.

Com o diagrama de caso de uso definido, foi gerado o modelo de domínio, representado pelo diagrama de domínio mostrado na Figura 5.

Figura 5



O objetivo do diagrama de domínio é representar visualmente as classes conceituais do mundo real, em um domínio de problema, e apresentar o entendimento de que informações o sistema tem e apresenta.

No diagrama de domínio mostrado na Figura 5, a Rede representa a rede de lojas, a entidade que possui as lojas e as agrupa. Rede também delimita a informação que o usuário pode ver, separando loja, meta, grupo, subgrupo, produto de outras redes.

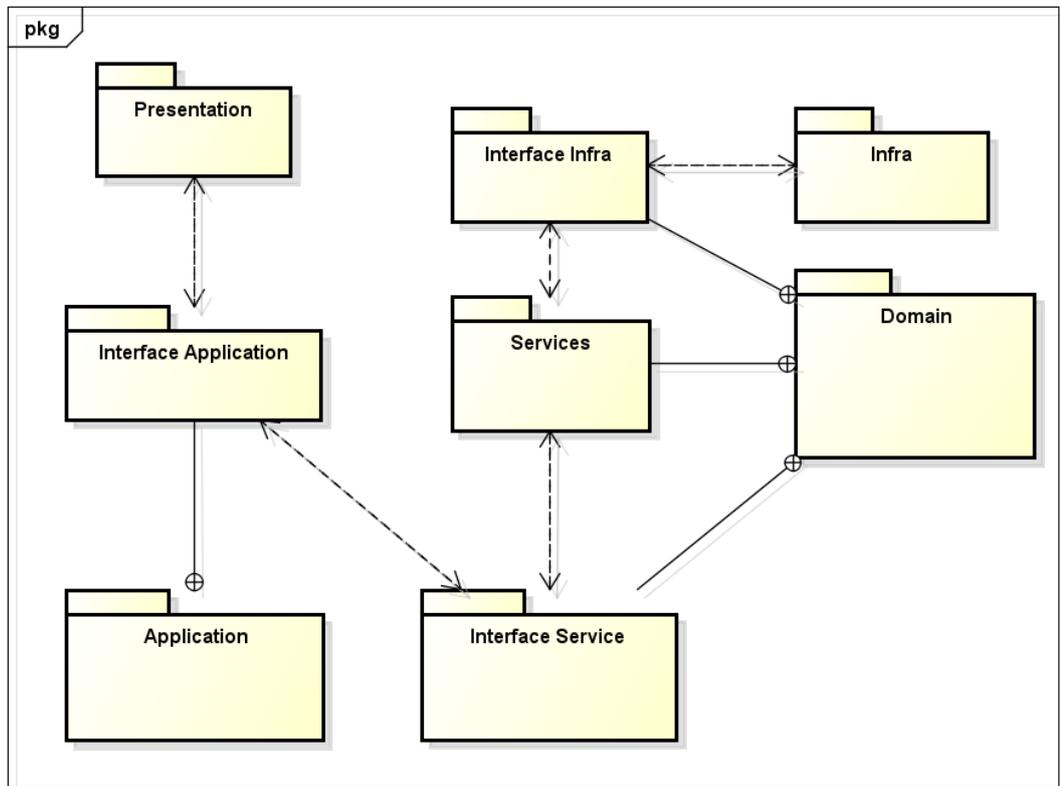
O conceito Meta representa o tipo de meta que pode ser determinada por vendedor, grupo, subgrupo. A Meta sempre estará vinculada a uma loja.

2.2.2 Projeto

A partir das informações obtidas na análise, foi realizado o projeto inicial, que se baseou no modelo de domínio e diagrama de classes. Com base nessas informações, foi definida a estrutura do projeto, escopo e padrão da interface do usuário.

Após a definição das classes necessárias no projeto, foi produzido o diagrama de pacotes, mostrado na Figura 6.

Figura 6



powered by Astah

O diagrama de pacotes demonstra como o sistema está dividido em módulos e as dependências entre eles. Foi adotada essa arquitetura para facilitar o reaproveitamento e possibilitar o baixo acoplamento entre as classes, permitindo que o sistema web e o sistema instalado na loja utilizem o mesmo domínio, reduzindo assim a quantidade de código gerada e mantendo os princípios do SOLID (HALL,2014).

2.2.3 Codificação

A linguagem utilizada foi C#, baseado na estrutura de DDD, utilizando o padrão de *Code First*. A plataforma de desenvolvimento (IDE) utilizada foi o Visual Studio 2015 Community da Microsoft. O sistema foi dividido em três partes: coleta de dados do banco de dados Linx, aplicação para cliente, aplicação no servidor. Foi utilizada a arquitetura de camadas na implementação do projeto. Para codificação do projeto, foram utilizados alguns frameworks,

como EntityFramework, AutoMapper, Ninject, entre outros. O banco de dados utilizado foi SQL Server 2014 Express.

O trecho de código fonte em C#, mostrado na Figura 7, é referente ao envio da informação para a API. O método *SendVenda* é responsável por coletar as vendas, no contexto da loja, e passar os objetos serializados para o método que se conecta a API. Já o método *SendVendaTask* é um método assíncrono que tem o conhecimento de como se comunicar com a API e os formatos que ela espera receber. Este método aguarda o retorno da API, e que, caso seja sucesso (HTTP 200 OK), retornará *verdadeiro* ou, em caso de falha, retornará *falso*.

Figura 7

```
public bool SendVenda(string instancia, string banco, string usuario, string senha, DateTime data, Loja loja)
{
    var vendasLoja = _iVendaService.GetVenda(instancia, banco, usuario, senha, data, loja);
    var json = JsonConvert.SerializeObject(vendasLoja);
    return SendVendaTask(json).Result;
}
private static async Task<bool> SendVendaTask(string json)
{
    using (var client = new HttpClient())
    {
        client.BaseAddress = new Uri("http://api.joaorcolombo.com");
        client.DefaultRequestHeaders.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));
        var request = new HttpRequestMessage(HttpMethod.Post, "v1/vendas")
        {
            Content = new StringContent(json, Encoding.UTF8, "application/json")
        };
        return await client.SendAsync(request).ContinueWith
            (responseTask => responseTask.Result.IsSuccessStatusCode);
    }
}
```

2.2.4 Teste

Os testes foram realizados em paralelo com o desenvolvimento e após a implantação. Foram utilizadas duas técnicas de testes: testes de unidade e teste de sistema. Os testes de unidade foram realizados nas principais funcionalidades do sistema e aplicado paralelamente com o desenvolvimento, já o teste de sistema foi realizado após a implementação.

Os testes de unidade foram realizados principalmente no domínio e também nas APIs, que recebem a informação da loja.

O teste, mostrado na Figura 8, é realizado na API de venda, que foi apresentada na Seção de codificação (2.2.3 Codificação). É redirecionado o repositório para um repositório de teste e é feito o envio das informações para a API. O principal objetivo deste teste é validar se a API está conseguindo receber os dados de venda com sucesso.

Figura 8

```
[TestClass]
public class VendaTest
{
    private IAppVendaService _iAppVendaService;

    [TestInitialize]
    public void Inicialize()
    {
        CompositionRoot.Wire(new NinjectFactory());
        _iAppVendaService = CompositionRoot.Resolve<IAppVendaService>();
    }
    [TestMethod]
    public void GetVenda_EnviaVendaApi_TRUE()
    {
        var conn = new DataConnection();
        var vendasLoja = _iAppVendaService.GetVenda
            (conn.Servidor, conn.Banco, conn.Usuario, conn.Senha, conn.Data, conn.Loja);
        var vendasController = new VendasController();
        var result = vendasController.PostVendas(vendasLoja);
        Assert.AreEqual(true, result.IsSuccessStatusCode);
    }
}
```

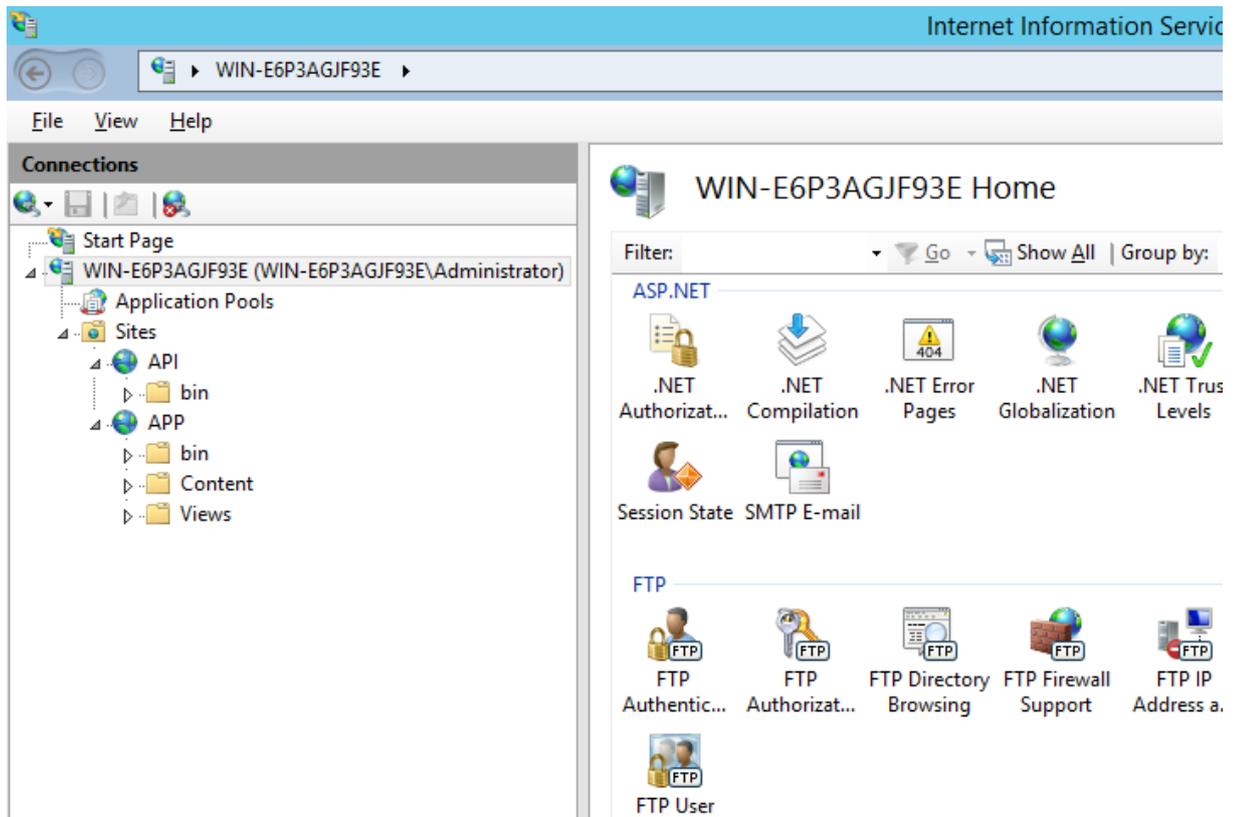
2.2.5 Implantação

A aplicação web utiliza o IIS8 (MICROSOFT) e é hospedada na Amazon. O aplicativo de envio de informação é instalado na loja e acessa os dados do banco de dados do Linx SQL Server 2008 R2, o mesmo envia os dados para o servidor web para seja, disponibilizados.

O IIS foi configurado com dois sites, um a para a aplicação (APP) e outro para as APIs (API) isso possibilita que os sites sejam gerenciados separadamente, possibilitando que por exemplo, somente as APIs migrem para um servidor mais robusto.

A Figura 9 mostra o menu de configuração do ISS na Amazon, com os dois sites criados, o APP e o API, para dar suporte a aplicação.

Figura 9



2.2.6 Manutenção

Os bugs são reportados por e-mail, sendo analisados nos quesitos, urgência, nível de dificuldade e impacto. Também é verificado se o erro é consequência de mau uso do usuário. Após o bug ser analisado, o analista retorna para o cliente informando se o bug será corrigido e o prazo para o ajuste. Para definir qual será a prioridade de atendimento, é atribuído um número de um a dez para os três quesitos e dividido por três: quanto maior o resultado maior a prioridade.

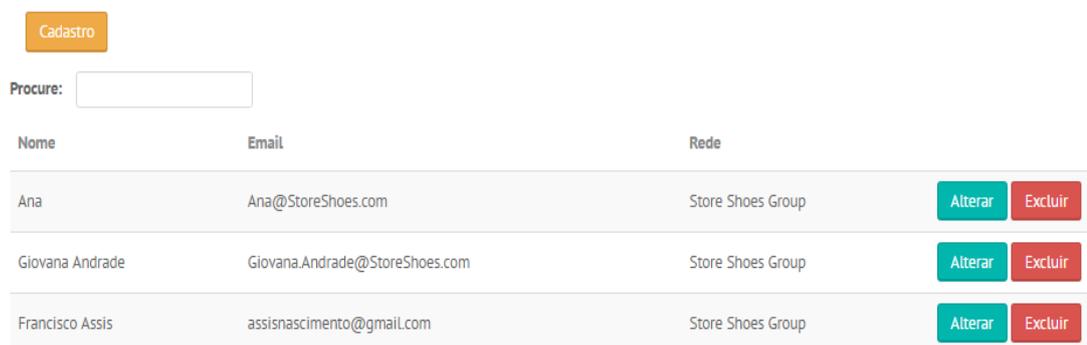
3 UTILIZAÇÃO DO SISTEMA

3.1 Funcionalidades para Lojista

O papel do lojista, no sistema desenvolvido neste trabalho, inicia no computador da loja. Ele pode definir uma data retroativa para o envio das informações para que o sistema, mesmo iniciando o uso na data do dia, já tenha as informações anteriores, podendo assim já começar a verificar os indicadores e estipular metas.

No sistema web, no segmento de usuários, ele pode fazer alterações no próprio cadastro ou nos usuários da rede que ele pertence, permitindo que seja alterada a senha, e-mail ou nome. Também é possível que ele cadastre mais usuários ou os exclua. (Conforme mostrado na Figura 10)

Figura 10



Nome	Email	Rede		
Ana	Ana@storeshoes.com	Store Shoes Group	Alterar	Excluir
Giovana Andrade	Giovana.Andrade@storeshoes.com	Store Shoes Group	Alterar	Excluir
Francisco Assis	assisnascimento@gmail.com	Store Shoes Group	Alterar	Excluir

No *dashboard*, são apresentados os indicadores padrões do varejo, gráficos de desempenho e informações pertinentes ao mês, porém também é possível incluir metas de maior importância para que seja facilitado o acesso e monitoramento das mesmas (conforme já mostrado anteriormente na Figura 3 e Figura 12).

As metas são divididas em duas partes: metas de cliente e metas de venda. O lojista pode cadastrar metas para as lojas pertencentes à rede, devendo informar um nome para identificar a meta; marcar “*exibe Dashboard*”, se deseja que seja exibida no *dashboard*; indicar uma data de início e fim; e também informar o valor a ser atingido.

Nas metas de venda, o lojista deve escolher uma medida que pode ser por valor de venda, quantidade de venda de produto ou número de vendas; e uma segunda medida, que pode ser por grupo, subgrupo e vendedor. Ao selecionar uma destas opções é apresentada uma listagem da opção; por exemplo, se for marcado vendedor será apresentada uma listagem dos vendedores pertencentes àquela loja, ficando por exemplo, “meta de venda para loja x, vendedor y/valor iniciando 01/01/2016 até 01/01/2017 com objetivo de atingir R\$50.000,00”. Após o cadastro da meta, o lojista é direcionado para o quadro de metas onde ele pode acompanhar o andamento das metas, visualiza-las com mais detalhes ou exclui-las. (Conforme mostrado na Figura 11)

Figura 11

Cadastro

Procure:

Nome	Progresso	%	Status	Data Final	Medida	Parametro	Loja		
Vendas Anual Calçado		4%	Em Progresso	31/12/2016	GrupoValor	CALÇADOS	Shoes SP1	Visualizar	Excluir
Venda Mesal Julho Calçados		3%	Encerrado	31/07/2016	GrupoValor	CALÇADOS	Shoes SP1	Visualizar	Excluir
Venda Anual Sapato Fechado Valor		8%	Em Progresso	31/12/2016	SubGrupoValor	FECHADO	Shoes SP1	Visualizar	Excluir
Bolsas 2 tri. Valor		39%	Em Progresso	31/12/2016	GrupoValor	BOLSAS	Shoes SP1	Visualizar	Excluir
Aumento Bolsa Setembro/Outubro Valor		92%	Em Progresso	31/10/2016	GrupoValor	BOLSAS	Shoes Norte	Visualizar	Excluir

3.2 Funcionalidades para Admin

O *Admin* executa somente as operações de CRUD (Create, Read, Update, Delete) no usuário e na rede por este motivo ele utiliza a interface do próprio SQL Manager.

3.3 Funcionalidades do Sistema

A principal funcionalidade do sistema é coletar os dados da loja diretamente do banco de dados de forma automática. O sistema verifica o campo “*DATA_PARA_TRANSFERENCIA*” disponível no sistema linx e compara com a data do último envio; caso a data seja maior, ele envia a informação. Os dados de conexão com o banco de dados e as informações necessárias para efetuar este processo ficam salvas nos registros do Windows dispensando a necessidade de um banco de dados próprio na loja. Este processo ocorre de 10 em 10 minutos, pois na instalação do aplicativo é criado um serviço no Windows para execução automática. Além do envio automático uma data pode ser selecionada para que seja enviado dados retroativos para loja.

4 RESULTADOS OBTIDOS

Para avaliar os resultados obtidos com o uso do sistema, foram submetidas mais de 2000 vendas, 1500 produtos e 2000 clientes de janeiro 2016 a setembro 2016 de duas diferentes lojas; e, no sistema, foram cadastradas 10 metas, 4 usuários e 5 lojas.

As informações foram enviadas ao longo de 3 meses. Em julho, foram submetidas as informações de janeiro a julho, onde estava concentrada a maioria das informações. Neste momento, foi verificado que deveria se aumentar o tempo de *TimeOut* da API para que o

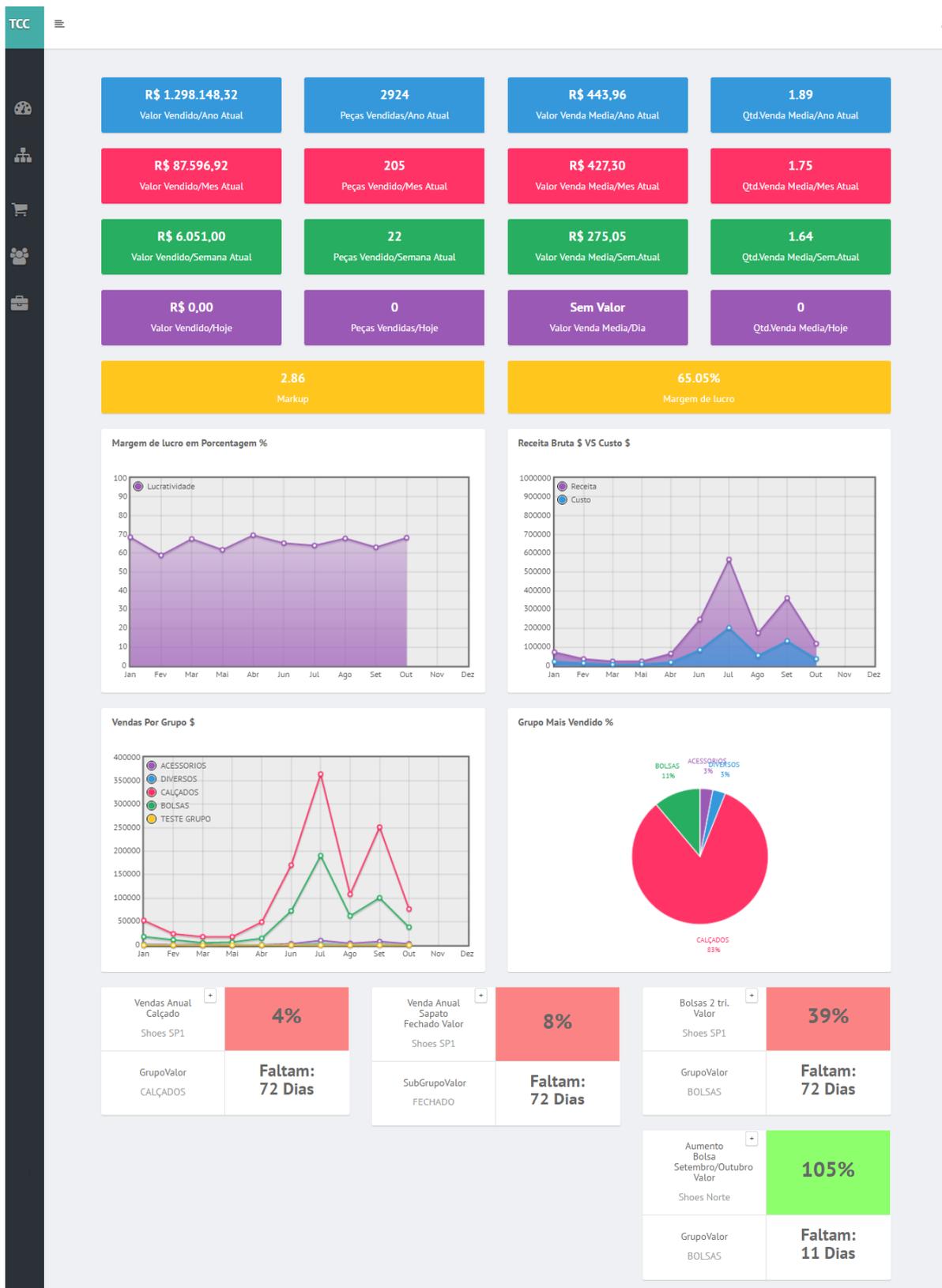
sistema da loja não perdesse a conexão com o sistema web. No mês de agosto e setembro, os envios de dados foram referentes ao período de cada mês, os envios foram feitos semanalmente demorando menos de 1 minuto para a sincronização da informação.

Com as informações no sistema web, foram cadastradas as metas para medir o desempenho. O processo de cadastro demorou cerca de 5 minutos, e foi realizado sem nenhum problema. Para realizar o cadastro, foram consultados os gestores de loja (colegas de empresa) que indicaram as metas que gostariam de acompanhar em suas lojas.

Para validação dos requisitos e expectativas em torno da ferramenta, foram consultados colegas na empresa em que o protótipo foi implantado, que são gestores de lojas e administradores de empresas. Todos repassaram um *feedback* positivo e com sugestões válidas para futuras implementações, tal como, coletar a informação de estoque para possibilitar mais controles e indicadores importantes.

Ao *logar* no sistema, o lojista verifica o *dashboard*, onde é apresentado um quadro geral da situação das lojas. (Conforme mostrado na Figura 12)

Figura 12



No *dashboard*, é possível o lojista verificar o andamento mensal da rede separado por vendedores e por loja, pode ser verificado a venda por grupo de produto e qual grupo de

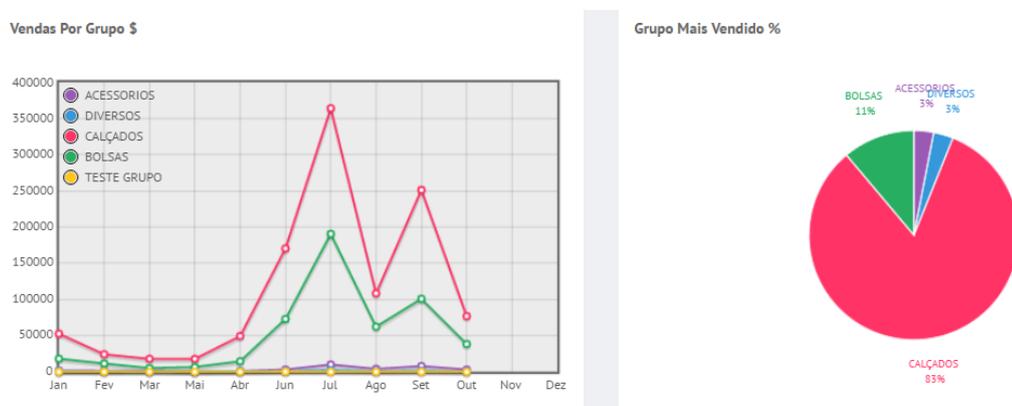
produto está sendo mais rentável, se a lucratividade está de acordo com o esperado, se o custo da mercadoria não está muito próximo ao preço de venda, entre outros indicadores.

Essas informações do *dashboard* fornecem o conhecimento necessário para que se possa responder perguntas, tais como, por exemplo:

a) Em que produtos preciso focar para atingir as metas?

Pode-se verificar os produtos mais rentáveis consultando Vendas Por Grupo e Grupo Mais Vendidos, assim será possível atingir metas de valor ou investir esforço nos grupos que não estão vendendo, para alcançar metas de quantidade. (Conforme mostrado na Figura 13)

Figura 13



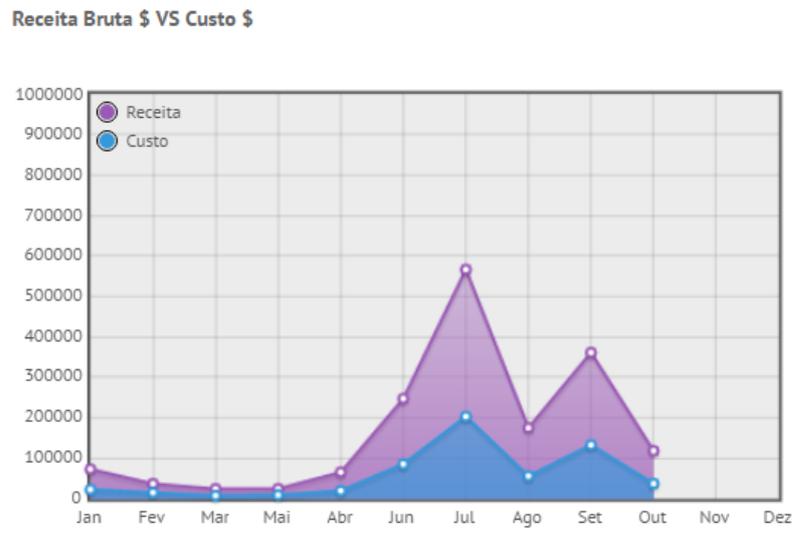
b) Qual grupo de produtos é indispensável para loja?

Pode-se consultar o gráfico de Vendas Por Grupo, e será possível ver os grupos que mais captam receita para a rede indicando que esses grupos são indispensáveis para o negócio prosperar. (Conforme mostrado também na Figura 13)

c) O custo e preço de venda do produto está correto?

Pode-se analisar o gráfico de Receita vs Custo: se as linhas estiverem muito próximas pode estar havendo um problema de custo ou de preço de venda. (Conforme mostrado na Figura 14)

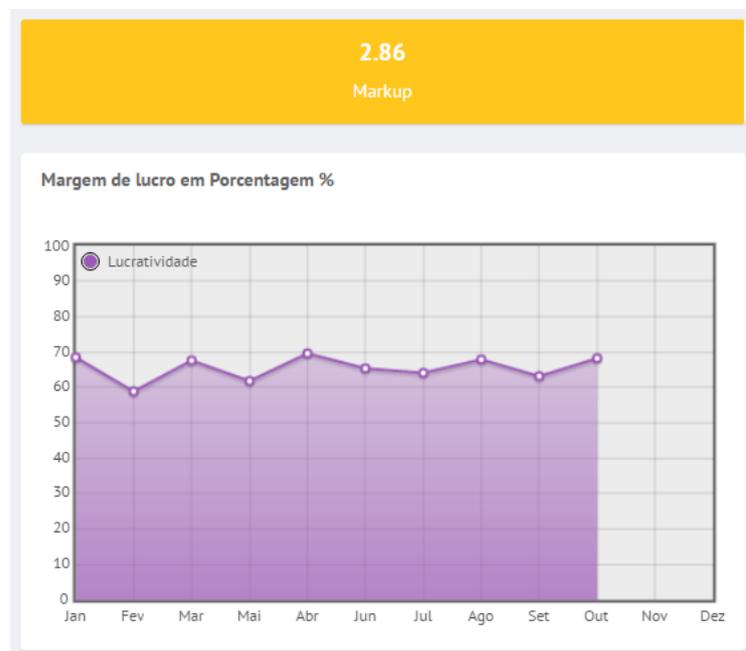
Figura 14



d) O markup está baixo?

Pode-se analisar markup e o gráfico de Lucratividade para descobrir se seu markup está diminuindo ou aumentando e verificar como está o markup atual no indicador de markup. (Conforme mostrado na Figura 15)

Figura 15



e) Preciso de mais vendedores?

Pode-se analisar a tabela de vendedores: se um vendedor tiver uma porcentagem muito grande na loja, é interessante pensar em mais vendedores.

f) Preciso de vendedores melhores?

Pode-se verificar a tabela de vendedores: se eles têm um percentual muito baixo na loja, talvez seja hora de um treinamento de venda.

g) Que loja precisa da minha atenção?

Pode-se verificar na parte inferior do dashboard a loja que está com o percentual menor no total da rede: esta loja pode estar com problemas.

h) Que metas precisam da minha atenção?

Pode-se verificar as metas que estão aparecendo no dashboard, essas metas estão em com a porcentagem em vermelho? Se sim, pode ser necessário dar uma olhada nelas, pois não estão evoluindo como deveriam.

5 CONCLUSÃO

Neste trabalho foi desenvolvido um sistema, onde é possível se acompanhar as vendas de uma rede de lojas de forma prática, fácil, integrada e online. O sistema foi desenvolvido com o intuito de fornecer as informações necessárias para que os lojistas ou gestores de rede de lojas tenham mais facilidade ao tomar decisões estratégicas.

Para aprimoramento futuro do sistema poderá ser coletada a informação sobre estoque da loja, permitindo o cálculo de estoque ideal referente a metas, cálculo de reserva de estoque, estoque de segurança, estoque sugerido, entre outros indicadores. Também será possível criar gráficos referentes ao estoque para melhor análise desta questão.

Caso haja oportunidade de expandir o sistema para outros sistemas base (além do Sistema Linx), o sistema será migrado para uma arquitetura Hexagonal onde será mais fácil lidar com os diferentes tipos de entrada e saída.

A ferramenta final atendeu às necessidades levantadas durante a análise do problema, auxiliando gestores e lojistas a tomar de decisões e ajudando-os a estipular metas para que seja possível medir o desempenho e projetar o crescimento esperado.

6 BIBLIOGRAFIA

TEIXEIRA, Juedir. **Consultor de Varejo**. Indicadores de desempenho no varejo. Disponível em http://www.juedirconsultor.com.br/index.php?option=com_content&task=view&id=24>. Acesso em 10 de fevereiro.

MICROSOFT,.Developer Network: C# Disponível em <<https://msdn.microsoft.com/pt-br/library/kx37x362.aspx> > Acesso 22 fevereiro.

EVANS, Eric. **Domain Driven Design –Tackling Complexity in the Heart of Software.** 1ºEd. 2016.

VERNON, Vaughn. **Implementando Domain Driven Design.** 1ºEd. 2005.

ALBAHARI, Joseph. **C# 4.0 Pocket Reference Instant Help for Programmers.** 3ºEd. 2010.

ALBAHARI, Ben. **C# Essentials.** 2ºEd.2010.

HALL, Gary McLean. **Adaptive Code via C#.** 1ºEd. 2014.

SATO, Alexandre. **Introdução ao ASP.NET Razor.** Disponível em <<https://msdn.microsoft.com/pt-br/library/gg675215.aspx> >. Acesso em 13 março.

MICROSOFT. **Visual Studio Community 2015.** Disponível em <<https://www.visualstudio.com/products/visual-studio-community-vs> > Acesso em 13 março.

MICROSOFT. **SQL Server 2014.** Disponível em < <https://www.microsoft.com/pt-br/server-cloud/products/sql-server/overview.aspx> > Acesso em 13 março.

MATTAR, FauzeNajib. **Administração de Varejo.** 1ºEd.2011.

PRIMAK, Fábio Vinícius. **Decisões com B.I.(Business Inteligence).**1ºEd.2008.

TARGIT. **Business Intelligence para LS Retail.** Disponível em <<http://www.targit.com/pt/solutions/by-industry/retail/ls-retail>> Acesso em 16 março.

ASCENT. **Business Intelligence.** Disponível em <<http://www.bi.grupoascent.com.br/>>Acesso em 16 março.

SANCHEZ, Fabrício. **Desenvolvimento web com ASP.NET MVC.**1ºEd. 2013.

CHAVES, Jair. **A Importância dos controles internos e evidenciação Das Informações Contábeis Frente A Tomada de Decisão.**1ºEd. 2008.

LORINO, Philippe. **O Economista e o Administrador.**3ª Ed.1992.

CRUZ, Fábio. **Scrum e Agile em Projetos Guia Completo.** 1ª ED. 2015.

MICROSOFT. Disponível em <<http://www.iis.net/>>. Acesso 17 março.

RODRIGUEZ, Martius. **Tecnologia de Informação e Gestão Empresarial.**2ªEd. 2004.