

# ESTUDO DE CASO

2023

## Definição da arquitetura de um sistema para Gestão dos Representantes

O objetivo deste documento é apresentar o estudo de caso da necessidade da empresa XYZ em construir um sistema de gestão para seus representantes.

## Estudo de Caso

### DEFINIÇÃO DA ARQUITETURA DE UM SISTEMA PARA GESTÃO DOS REPRESENTANTES

### APRESENTAÇÃO DA EMPRESA XYZ

A empresa XYZ, com sede situada na cidade de Joinville, atua como revenda de componentes eletrônicos para os três estados do sul do Brasil: Rio Grande do Sul, Santa Catarina e Paraná, onde possui uma filial em cada. Seu principal mercado está na revenda de componentes eletrônicos importados de países como China e Estados Unidos para grandes empresas também a venda para pessoas físicas. Hoje a empresa conta com dois sistemas que fazem a gestão da informação: um ERP e um e-commerce. A empresa XYZ conta com os seguintes setores na sua sede: diretoria, administrativo/contábil, compras, estoque, vendas e tecnologia da informação, nas demais filiais existe os setores: administrativo, estoque e vendas. Hoje a empresa possui apenas um setor de vendas interno em cada loja que cuida de vendas diretas para empresas

ou vendas no balcão, o objetivo é estruturar uma equipe de vendas que faça visitas a potenciais clientes, apresente novos produtos e realize o fechamento de novas vendas.

## OBJETIVO

A empresa tem como objetivo o desenvolvimento de um sistema para gestão das três equipes de vendas.

## PREMISSAS DO PROJETO

- O novo sistema deverá:
  - ter integrações com os sistemas legados da empresa;
    - dados dos funcionários
    - dados dos produtos
    - situação de estoque dos produtos
    - integração com o setor de vendas
    - emissão de documentos de cobrança financeira
    - pagamento de comissão para os vendedores
  - permitir a consulta ao catálogo de produtos disponível nas três lojas;
  - permitir uma visão geral do desempenho dos representantes;
    - registro de potenciais clientes
    - registro da visita aos clientes
    - registro de pedidos de venda durante a visita aos clientes
  - calcular os valores de comissão por vendas;
  - acompanhar os pedidos de venda realizados;
  - deverá ser acessado de qualquer local com internet
- O sistema de e-commerce:
  - permitir que os clientes acompanhem os pedidos realizados pelos representantes;
- O sistema ERP
  - Prove dados sobre os funcionários
  - Prover a listagem de produtos
  - Prover a situação de estoque em todas as filiais
  - Permitir a integração de pedidos de venda em todas as filiais
  - Permitir o acompanhamento dos pedidos
  - Permitir a geração de documentos financeiros de cobrança
  - Permitir a geração de pagamento das comissões dos representantes

## SISTEMA PRETENDIDO

### Atores primários do sistema

- Funcionário administrador
- Funcionário representante

### Atores secundários

- Sistema ERP
- Sistema e-commerce

### Casos de uso do sistema de gestão de representantes

- Manter clientes
- Criar prospecções
- Registrar Visita
- Consultar funil de vendas
- Sincronizar lista de produtos
- Consultar produtos
- Realizar pedido

## SISTEMAS LEGADOS – ERP



**Gestão de Pessoas:** módulo responsável pelo registro de contratação dos funcionários, pela emissão da folha de pagamento.

**Financeiro:** módulo responsável pelo setor de contas a pagar, contas a receber.

**Estoque:** módulo responsável pelo controle do número de itens de um determinado produto.

**Vendas:** módulo responsável pelo registro de vendas, geração dos processos de cobrança, realização do processo de envio dos produtos para os clientes.

**Compras:** módulo responsável pelo cadastro de produtos, registro de fornecedores, compra de produtos para reposição dos estoques.

**Integração:** módulo responsável por permitir que os dados de todos os setores possam ser integrados a sistemas externos através de webservices no padrão RESTful.

O sistema ERP foi totalmente construído em C# .net e ASP.net MVC, utiliza banco de dados relacional, e possui uma camada de integração que dá acesso aos dados pela API REST. A arquitetura física da aplicação esta organizada conforme a figura 1, a solução esta distribuída em dois conjuntos de servidores não virtualizados, onde um servidor de aplicação e um servidor de banco de dados dão suporte a todo o ERP para todas as filiais, e um outro conjunto de servidor de aplicação e de banco de dados da suporte ao e-commerce. Ambos os conjuntos de servidores ficam em uma estrutura de data center terceirizada, que provê tanto a conectividade quanto a questão de segurança da informação.

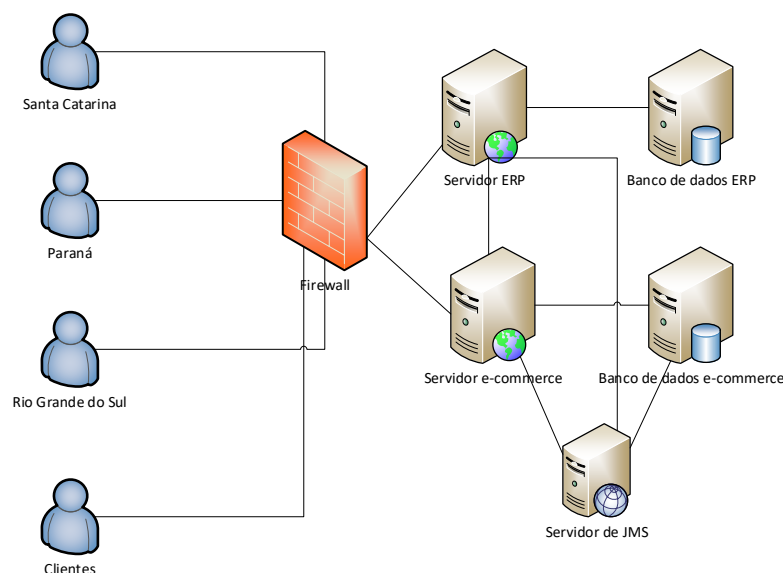
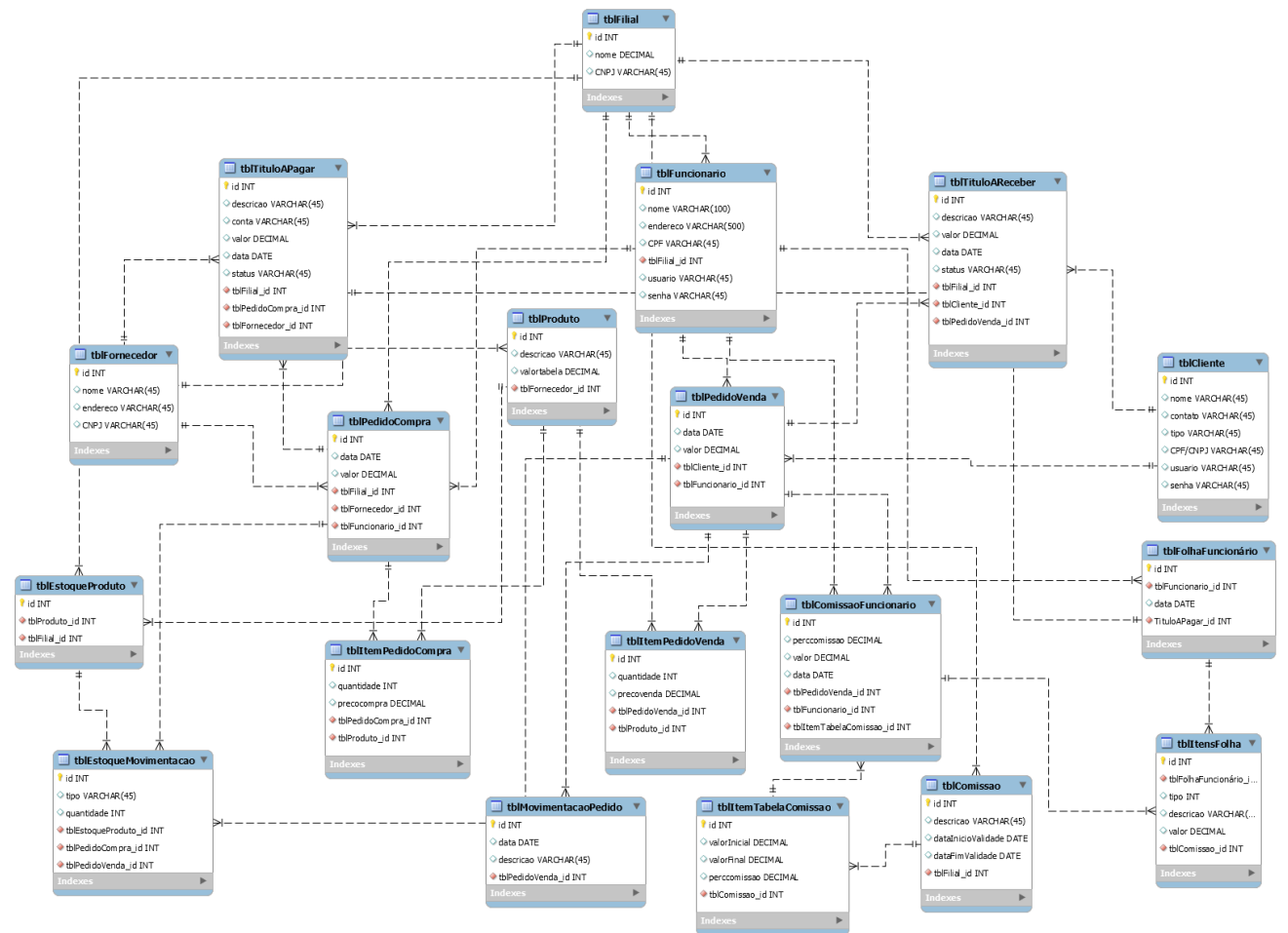


FIGURA 1 - ARQUITETURA FÍSICA

## Modelo de entidades e relacionamentos do ERP



## Documentação das APIs RESTful

### Autorização

Método que permite a autenticação de usuários.

GET

```
/api/auth
```

### Parâmetro

Campo	Tipo	Descrição
username	String	Nome do usuário
Password	String	Senha do usuário

**Success 200**

Campo	Tipo	Descrição
success	Boolean	Status da requisição
data	JSON	Objeto contendo token de autenticação

```
HTTP/1.1 200 OK
{
  "success": true,
  "data": {
    "token":
    "eyJ0eXAiOiIzI1NiJ9.eyJ1c2VyIHd1cHcuME1vSGFQR1zF9.awGvx_C7O5TCdT6F2bPn
    xQ4frLVPqdXWGJlmd7S_e1Q"
  }
}
```

**Listagem de Produto**

Método que permite buscar a lista de produtos

GET

/api/products

**Parâmetro**

Campo	Tipo	Descrição
token	String	token de autorização

**Success 200**

Campo	Tipo	Descrição
success	Boolean	Status da requisição
data	JSON	Lista de objetos com os dados do produto

```
HTTP/1.1 200 OK
{
  "success": true,
  "data": [
    {
      "id": 1234,
      "descricao": "resistor 220k ohm",
      "preco venda": 1.0,
      "fornecedor": {
        "nome": "distribuidor x"
      },
      "stock_qtd": 1000
    }
  ]
}
```

**Produto por descrição**

Método que permite buscar dados de um produto pela descrição

GET

`/api/products/desc`**Parâmetro**

Campo	Tipo	Descrição
token	String	token de autorização
Description	String	Texto com a descrição do produto

**Success 200**

Campo	Tipo	Descrição
success	Boolean	Status da requisição
data	JSON	Objeto com os dados do produto

```
HTTP/1.1 200 OK
{
  "success": true,
  "data": {
    "id": 1234,
    "descricao": "resistor 220k ohm",
    "preco venda": 1.0,
    "fornecedor": {
      "nome": "distribuidor x"
    },
    "stock_qtd": 1000
  }
}
```

**Produto por fornecedor**

Método que permite buscar dados de um produto pelo nome do fornecedor

GET

`/api/products/supplier`**Parâmetro**

Campo	Tipo	Descrição
token	String	token de autorização
description	String	Texto com a descrição do fornecedor



**Success 200**

Campo	Tipo	Descrição
success	Boolean	Status da requisição
data	JSON	Objeto com os dados do produto

```
HTTP/1.1 200 OK
{
  "success": true,
  "data": {
    "id": 1234,
    "descricao": "resistor 220k ohm",
    "preco venda": 1.0,
    "fornecedor": {
      "nome": "distribuidor x"
    },
    "stock_qtd": 1000
  }
}
```

**Registrar Cliente**

Método que permite cadastrar um novo cliente

**POST****/api/clients****Parâmetro**

Campo	Tipo	Descrição
token	String	token de autorização
Data	JSON	Objeto contendo os campos necessários para registrar um cliente: nome, contato, tipo, CPF/CNPJ, login, senha

**Success 200**

Campo	Tipo	Descrição
success	Boolean	Status da requisição

```
HTTP/1.1 200 OK
{
  "success": true
}
```

**Atualizar Cliente**

Método que permite atualizar os dados de um Cliente

**PUT****/api/clients****Parâmetro**

Campo	Tipo	Descrição
token	String	token de autorização
Data	JSON	Objeto contendo os campos necessários para atualizar um cliente: id, nome, contato, tipo, CPF/CNPJ, login, senha

**Success 200**

Campo	Tipo	Descrição
success	Boolean	Status da requisição

```
HTTP/1.1 200 OK
{
  "success": true
}
```

### Buscar Cliente

Método que permite buscar os dados de um cliente por CPF/CNPJ

GET

/api/clients

#### Parâmetro

Campo	Tipo	Descrição
token	String	token de autorização
Data	String	CPF/CNPJ do cliente

#### Success 200

Campo	Tipo	Descrição
success	Boolean	Status da requisição
Data	JSON	Objeto com os dados do cliente

```
HTTP/1.1 200 OK
{
  "success": true,
  "data": {
    "id": 123,
    "nome": "Antonio",
    "contato": "Rua lalaal 200 Telefone 55551234",
    "tipo": "fisica",
    "CPFCNPJ": 1234567890123
  }
}
```

## Fazer pedido de venda

Método que permite incluir um pedido de venda

POST

/api/sells

### Parâmetro

Campo	Tipo	Descrição
token	String	token de autorização
Data	JSON	Objeto contendo os dados de um pedido de venda.

```
{
  "data": 2012-04-23T18:25:43.511Z,
  "id_client": 123,
  "products": [
    { "id": 1000,
      "qtd": 2,
      "price": 2.6 },
    { "id": 1001,
      "qtd": 6,
      "price": 3.4 }
  ],
  "price_total": 25.6,
  "payment_method": 1
}
```

### Success 200

Campo	Tipo	Descrição
success	Boolean	Status da requisição

```
HTTP/1.1 200 OK
{
  "success": true
}
```

## Lista pedidos de um cliente

Método que permite listas os pedidos de um cliente

GET

/api/sells

### Parâmetro

Campo	Tipo	Descrição
token	String	token de autorização
Data	JSON	Objeto contendo o ID do cliente

### Success 200

Campo	Tipo	Descrição
success	Boolean	Status da requisição
Data	JSON	Lista com os dados do pedido de um cliente

```
HTTP/1.1 200 OK
{
  "success": true,
  "data": [
    {
      "data": 2012-04-23T18:25:43.511Z,
      "id_client": 123,
      "products": [
        { "id": 1000,
          "qtd": 2,
          "price": 2.6 },
        { "id": 1001,
          "qtd": 6,
          "price": 3.4 }
      ],
      "price_total": 25.6,
      "payment_method": 1,
      "delivery": [
        { "data": 2012-04-23T18:25:43.511Z,
          "status": "for immediate delivery"
        }
      ]
    }
  ]
}
```

## SISTEMAS LEGADOS – E-COMMERCE



**Vitrine de produtos:** módulo responsável por apresentar os produtos para os clientes que visitam o site da empresa, além de disponibilizar informações institucionais.

**Área do Cliente:** módulo responsável pela criação de novas contas dos cliente, alteração de senha e consulta do histórico de compras.

**Rotina de venda:** módulo responsável por registrar os itens que os clientes desejam comprar e criar os pedidos de venda no ERP.

**Rastreamento:** módulo responsável por consultar o status das movimentações de entrega dos produtos da venda realizada.

**Administração do Site:** módulo responsável pelas integrações com o ERP, gerenciamento dos

conteúdos exclusivos do site.

O software e-commerce foi desenvolvido na linguagem JAVA e se utiliza de um servidor de banco de dados relacional para armazenar as informações. Existem rotinas de integração com o ERP da empresa, que precisam ser complementadas pois nem todas as informações dos produtos estão no banco de dados do ERP, ex: (foto, links para manuais, etc). A integração acontece diariamente através de um agendamento que dispara consultas para o webservice do ERP, onde são sincronizados principalmente os dados dos produtos. Devido o grande volume de acessos ao site, a rotina de vendas não integra diretamente os pedidos de venda com o ERP. Ela utiliza um mecanismo de fila de mensagens JMS, toda vez que uma compra é realizada no site, o sistema e-commerce gera uma mensagem JMS que é transmitida para o servidor, e essa mensagem é enfileirada pelo mecanismo. Com isso ela passa por diversos estágios, dentre eles: confirmação do estoque do produto, aprovação do processo de pagamento, então o pedido é registrado no ERP.

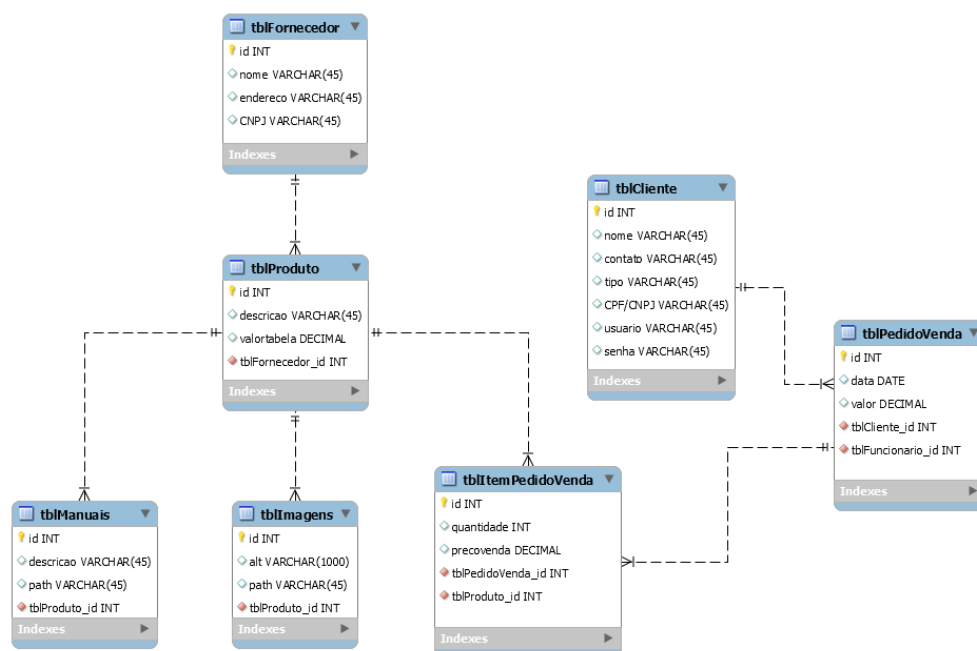


FIGURA 2 - MER E-COMMERCE