

Documentação do Sistema

Amanda Marino
Gustavo Meira
Gustavo Kravensko
Gabriel Rodrigues
Neemias Borges

14 de novembro de 2024

Sumário

| | | |
|-----------|--|-----------|
| 1 | Introdução | 3 |
| 1.1 | Contexto | 3 |
| 2 | Problema Identificado | 3 |
| 3 | Solução Proposta | 3 |
| 3.1 | Gamificação do Cadastro | 3 |
| 3.2 | Automatização via APIs | 3 |
| 4 | Endpoints utilizados no Sistema | 3 |
| 4.1 | Consulta de CEP | 3 |
| 4.2 | Consulta CNPJ | 3 |
| 5 | Requisitos do Sistema | 3 |
| 5.1 | Geração de Código do Cliente | 3 |
| 5.2 | Campos do Cadastro | 4 |
| 6 | Funcionalidades | 4 |
| 6.1 | Tela de Listagem | 4 |
| 6.2 | Tela de Edição | 5 |
| 6.3 | Exclusão de Cliente | 5 |
| 6.4 | Notificações por E-mail | 5 |
| 7 | Parâmetros de Consulta | 5 |
| 7.1 | Listagem de Clientes | 5 |
| 8 | Estruturas de Dados | 6 |
| 8.1 | Retorno da API da Receita | 6 |
| 9 | Tecnologias Utilizadas | 6 |
| 10 | Documentação dos Endpoints da API em TLPP | 7 |
| 11 | Códigos de Resposta | 10 |
| 12 | Referências | 10 |

1 Introdução

1.1 Contexto

Sistema Web Desenvolvido para prestador de serviço no ramo de Móveis Planejados, visando melhorar a forma de cadastros do Objeto de Cliente dentro do sistema TOTVS já implementado.

2 Problema Identificado

- Complexidade no cadastro de clientes devido ao excesso de campos

3 Solução Proposta

3.1 Gamificação do Cadastro

- Implementação de vertical stepper para organização dos campos em blocos
- Checkpoints de preenchimento para melhor experiência do usuário

3.2 Automatização via APIs

- Integração com API da Receita Federal (WS)
- Integração com API dos Correios
- Integração com API do Protheus

4 Endpoints utilizados no Sistema

4.1 Consulta de CEP

```
GET https://viacep.com.br/ws/{cep}/json/
```

4.2 Consulta CNPJ

```
GET https://receitaws.com.br/v1/cnpj/{cnpj}
```

5 Requisitos do Sistema

5.1 Geração de Código do Cliente

- Prefixo 'CL' seguido por sequência de 3 dígitos numéricos

5.2 Campos do Cadastro

- a) Código do Cliente
- b) Loja
- c) Nome
- d) Física/Jurídica
- e) Endereço
- f) Nome Fantasia
- g) Bairro
- h) Tipo
- i) Estado
- j) Código do Município
- k) CEP
- l) Município
- m) CNPJ/CPF
- n) DDD
- o) Telefone
- p) País
- q) Email
- r) Home-page
- s) Data Abertura/Nascimento
- t) Status
- u) Complemento

6 Funcionalidades

6.1 Tela de Listagem

- Filtros por:
 - Código
 - Nome/ Razão Social
 - CPF ou CNPJ
 - Tipo do Cliente
 - Estado vinculado
- Paginação
- Filtragem em tempo real

6.2 Tela de Edição

- Campos não editáveis:
 - Código
 - CPF ou CNPJ
 - Tipo do Cliente
- Campos Editáveis para Pessoa Física:
 - Nome
 - Dados do Endereço
 - Data de Vinculação
 - Dados de Contato
- Campos editáveis para Pessoa Jurídica:
 - Nome
 - Dados do Endereço
 - Data de Vinculação
 - Dados de Contato

6.3 Exclusão de Cliente

- Modal de confirmação de exclusão

6.4 Notificações por E-mail

- Notificação automática após alteração de registro
- Um único usuário receberá as notificações
- E-mail configurável
- Resumo de todos os campos editados (exceto endereço)

7 Parâmetros de Consulta

7.1 Listagem de Clientes

- **show** (int32)
 - Número de itens na página de resultado
- **page** (int32)
 - Número da página
- **with** (string)
 - Obter recursos com relações

8 Estruturas de Dados

8.1 Retorno da API da Receita

```
1 {
2   "abertura": "string",
3   "situacao": "string",
4   "tipo": "string",
5   "nome": "string",
6   "fantasia": "string",
7   "porte": "string",
8   "natureza_juridica": "string",
9   "qsa": [],
10  "logradouro": "string",
11  "numero": "string",
12  "municipio": "string",
13  "bairro": "string",
14  "uf": "string",
15  "cep": "string",
16  "email": "string",
17  "telefone": "string",
18  "data_situacao": "string",
19  "motivo_situacao": "string",
20  "cnpj": "string",
21  "ultima_atualizacao": "string",
22  "status": "string",
23  "complemento": "string",
24  "efr": "string",
25  "situacao_especial": "string",
26  "data_situacao_especial": "string",
27  "atividade_principal": [],
28  "atividades_secundarias": [],
29  "capital_social": "string",
30  "simples": {},
31  "simei": {},
32  "extra": {},
33  "billing": {}
34 }
```

9 Tecnologias Utilizadas

- Frontend:
 - Angular
 - Materialize
- Backend:
 - Node.js
 - TLPP

10 Documentação dos Endpoints da API em TLPP

POST /v1/clientes/incluir

Descrição: Método para inserir um novo cliente no sistema, registrando suas informações básicas e vinculando ao sistema CRM.

Parâmetros:

Não possui parâmetros de path

Responses:

200 - Sucesso

404 - Não Encontrado

GET /v1/clientes/listar

Descrição: Método para listar todos os clientes cadastrados no sistema, retornando uma lista com informações detalhadas de cada cliente.

Parâmetros:

Não possui parâmetros de path

Responses:

200 - Sucesso

404 - Não Encontrado

GET /v1/clientes/:codigo

Descrição: Método para obter detalhes de um cliente específico com base em seu código. Retorna informações completas do cliente selecionado.

Parâmetros:

- **codigo** (int32) - ID do cliente *[required]*

Responses:

200 - Sucesso

404 - Cliente Não Encontrado

PUT /v1/clientes/atualizar/:codigo/:loja

Descrição: Método para atualizar as informações de um cliente específico, incluindo detalhes de contato e preferências, com base no código e na loja.

Parâmetros:

- **codigo** (int32) - ID do cliente *[required]*
- **loja** (int32) - ID da loja *[required]*

Responses:

200 - Sucesso

404 - Cliente ou Loja Não Encontrado

DELETE /v1/clientes/deletar/:codigo/:loja

Descrição: Método para excluir um cliente do sistema com base em seu código e loja, removendo todas as informações associadas ao cliente.

Parâmetros:

- **codigo** (int32) - ID do cliente *[required]*
- **loja** (int32) - ID da loja *[required]*

Responses:

200 - Sucesso

404 - Cliente ou Loja Não Encontrado

GET /v1/clientes/ultimoCodigo

Descrição: Método para recuperar o último código de cliente registrado no sistema, auxiliando no controle de novos cadastros.

Parâmetros:

Não possui parâmetros de path

Responses:

200 - Sucesso

404 - Não Encontrado

GET /v1/clientes/municipiosporestado/:estado

Descrição: Método para listar todos os municípios de um estado específico, retornando uma lista de municípios que pertencem ao estado indicado.

Parâmetros:

- **estado** (string) - Sigla do estado *[required]*

Responses:

200 - Sucesso

404 - Estado Não Encontrado

GET /v1/clientes/codigomunicipio/:estado/:municipio

Descrição: Método para obter o código de um município específico, usando o estado e o nome do município como parâmetros de entrada.

Parâmetros:

- **estado** (string) - Sigla do estado *[required]*
- **municipio** (string) - Nome do município *[required]*

Responses:

200 - Sucesso

404 - Estado ou Município Não Encontrado

11 Códigos de Resposta

- **200 OK:** Sucesso na operação.
- **204 No Content:** Sucesso, mas sem conteúdo no corpo da resposta.
- **404 Not Found:** Recurso não encontrado.
- **500 Internal Server Error:** Erro no servidor ao processar a requisição.
- **503 Service Unavailable:** Serviço temporariamente indisponível.

12 Referências

- TOTVS Protheus API Documentation.
- Receita Federal API Documentation.
- ViaCEP API Documentation.