**UNIVERSIDADE ANHEMBI MORUMBI**

**UNIDADE PAULISTA I**

**Desenvolvimento de Sistemas**

GUILHERME MENDES NASCIMENTO BERNARDO

125221101237

**SÃO PAULO**

**2021**

GUILHERME MENDES NASCIMENTO BERNARDO

Monografia apresentada à Universidade Anhembi Morumbi – Paulista I como requisito parcial à conclusão do semestre, sob orientação dos professores(a) Rebecca Bignardi Arambasic da matéria de Programação de soluções computacionais e Andreza Benatti Cassetarri da matéria de Modelagem de Software.

**SÃO PAULO**

**2021**

GUILHERME MENDES NASCIMENTO BERNARDO

São Paulo, 2021

**RESUMO**

Muitas empresas tinham o costume de guardar o cadastro de seus clientes em fichas de papel, porém, isso gera diversos problemas, como o grande espaço necessário para guardar uma grande quantidade de fichas, a fragilidade das informações nas fichas e o alto tempo necessário para cadastrar novas fichas e organizar os arquivos já existentes. Este aplicativo gera uma solução para o armazenamento das fichas e introduz um novo método de cadastro automático das fichas. A *API* utilizada gera dados que são utilizados em cadastro de clientes de forma automática.

Palavras-chave: música, músicos em ascensão, web marketing.

**SUMÁRIO**

[1 INTRODUÇÃO 5](#_Toc121328714)

[1.1 PROBLEMATIZAÇÃO 5](#_Toc121328715)

[1.2 JUSTIFICATIVA 5](#_Toc121328716)

[1.3 OBJETIVO GERAL 5](#_Toc121328717)

[1.4 OBJETIVOS ESPECÍFICOS 6](#_Toc121328718)

[2 DESENVOLVIMENTO 6](#_Toc121328719)

[2.1 SOFTWARES 6](#_Toc121328720)

[2.2 CLASSES E METODOS 6](#_Toc121328721)

[2.3 BANCO DE DADOS 8](#_Toc121328722)

[2.3.1 MOVIMENTAÇÕES DE DADOS 8](#_Toc121328723)

[2.4 DIAGRAMAS 9](#_Toc121328724)

[2.4.1 CASO DE USO 9](#_Toc121328725)

[2.4.2 CLASSE 10](#_Toc121328726)

[2.4.3 ATIVIDADES 10](#_Toc121328727)

[2.4.4 SEQUÊNCIA 10](#_Toc121328728)

[2.4.5 MER 11](#_Toc121328729)

[2.8 REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS 12](#_Toc121328730)

[2.8.1 REQUISITOS FUNCIONAIS 12](#_Toc121328731)

[2.8.2 REQUISITOS NÃO FUNCIONAIS 12](#_Toc121328732)

[3 CONCLUSÃO 12](#_Toc121328733)

[4 REFERÊNCIAS 13](#_Toc121328734)

# 1 INTRODUÇÃO

## 1.1 PROBLEMATIZAÇÃO

Muitas empresas tinham o costume de guardar o cadastro de seus clientes em fichas de papel, porém, isso gera diversos problemas, como o grande espaço necessário para guardar uma grande quantidade de fichas, a fragilidade das informações nas fichas e o alto tempo necessário para cadastrar novas fichas e organizar os arquivos já existentes.

Este aplicativo gera uma solução para o armazenamento das fichas e introduz um novo método de cadastro automático das fichas, e *API* utilizada gera dados que são utilizados em cadastro de clientes de forma automática.

## 1.2 JUSTIFICATIVA

O método de guardar fichas de cadastro é muito obsoleto, além de ser algo perigoso em relação as perdas de dados. Muitas empresas que já estão no mercado tem a tendencia a se informatizar, isso é visto no dia a dia, um exemplo seria o dono de mercearia ou um dono de quitanda, que em tempos mais recentes utilizam cada vez mais a tecnologia para o gerenciamento de seu comercio. Este aplicativo seria mais uma maneira de ajudar a informatização.

## 1.3 OBJETIVO GERAL

Armazenar dados de cadastro de clientes e forma informatizada, e gerar novos cadastros.

## 1.4 OBJETIVOS ESPECÍFICOS

Para se atingir o nosso objetivo geral, foram criados objetivos específicos para uma melhoria no sentido de praticidade e foco ao nosso objetivo geral, onde podemos destacar:

* Integrar a API de consultas por CNPJ ao aplicativo principal.
* Criar uma interface simples e intuitiva para o usuário com uma tabela para exibição dos dados e uma tela de edição dos cadastros já existentes.
* Criar um banco capaz de suprir as necessidades do aplicativo.

# 2 DESENVOLVIMENTO

## 2.1 SOFTWARES

Para produção de documentos foi utilizado Microsoft Word, para a produção das apresentações usamos o Microsoft PowerPoint juntamente com o *OBS* para a gravação da apresentação, para a criação do banco de dados foi utilizado a linguagem *MySQL* e como *SGBD* foi utilizado o *PHPMyAdmin*, a Criação dos diagramas foi feita com o *Dia Diagram*, e todo o código foi produzido com o eclipse juntamente com as *libraries* Swing(Criação de Interface), *WindowBuilder*(GUI para criação de interfaces gráficas), *GSON*(Formatação de String JSON), *JSoup*(realiza a coleta de dados na web).

## 2.2 CLASSES E METODOS

No aplicativo temos 5 classes sendo 2 delas interfaces gráficas, 1 classe auxiliar conhecida como *BEAN*, 1 classe para as ações do banco e a última para a consulta na web.

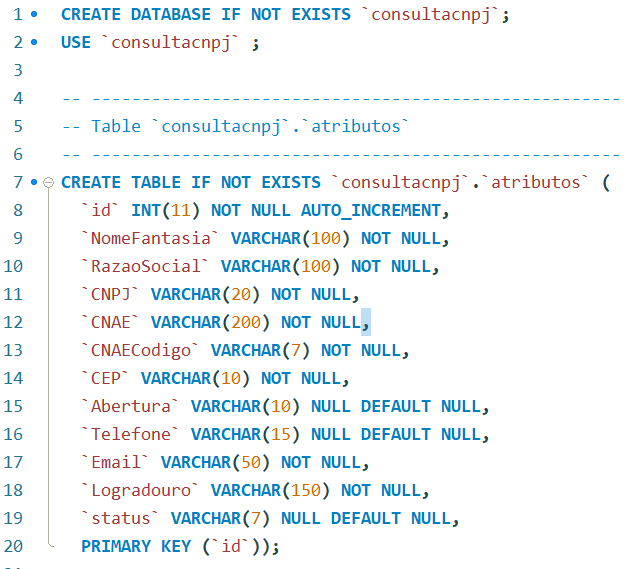
A primeira classe de interface gráfica é a responsável por mostrar todas a fichas cadastradas, além de possibilitar a entrada de novas fichas a partir do CNPJ informado, também é possível acessar um menu *POPUP* apertando o botão direito em cima de uma linha da tabela, lá é possível deletar tal linha, ou acessar a segunda interface, que é a responsável pela edição dos dados, lá os dados da tabela principal são expandidos e é disponibilizado a opção de editar os dados ou os deletar.

A classe auxiliar *BEAN* é uma classe muito utilizada em *JAVA*, ela é uma classe para a transmissão de informação entre classes. O *BEAN* é composto de variáveis privadas e métodos get e set para as respectivas variáveis.

A classe do banco é padrão com método de conexão com o banco, uma *STRING* de conexão além dos métodos padrões de *INSERT*, *DELETE*, *UPDATE* e *READ*, porém na minha classe tem um método além dos padrões, que é o método de *READCNPJ* que é um método utilizado pela classe de edição das fichas citada no segundo parágrafo.

Por fim a classe de consulta na web se comunica com a api, retorna os dados e os formata até virarem *JSON* que pode ser utilizado com a *library* *GSON*, depois são extraídos do *JSON* e armazenados em *ARRAYS*.

## 2.3 BANCO DE DADOS



### 2.3.1 MOVIMENTAÇÕES DE DADOS

#### Inserção de dados

public static void insert (String NomeFantasia, String RazaoSocial, String CNPJ, String CNAE, String CNAECodigo, String CEP, String Abertura, String Telefone, String Email, String Logradouro, String status)

{

String sql = "INSERT INTO `atributos` (`NomeFantasia`, `RazaoSocial`, `CNPJ`, `CNAE`, `CNAECodigo`,

`CEP`, `Abertura`, `Telefone`, `Email`, `Logradouro`, `status`) "+ "VALUES ('" + NomeFantasia

+ "','" + RazaoSocial + "','" + CNPJ + "','" + CNAE + "','" + CNAECodigo + "','" + CEP + "','"

+ Abertura + "','" + Telefone + "','" + Email + "','" + Logradouro + "','" + status + "');";

*exec*(sql);

}

#### Exclusão de dados

public static void delete (String id)

{

String sql = "DELETE FROM `atributos` WHERE CNPJ ='" + id + "';";

*exec*(sql);

}

#### Edição de dados

public static void update(String NomeFantasia, String RazaoSocial, String CNPJ, String CNAE, String CNAECodigo, String CEP, String Abertura, String Telefone, String Email, String Logradouro)

{

String sql = "UPDATE atributos " + "SET NomeFantasia = '" + NomeFantasia + "', RazaoSocial = '" +

RazaoSocial + "', CNPJ ='" + CNPJ + "', CNAE = '" +CNAE + "', CNAECodigo = '" + CNAECodigo

+ "', CEP = '" + CEP + "', Abertura = '" + Abertura + "', Telefone = '" + Telefone

+ "', Email = '" + Email + "', Logradouro = '" + Logradouro + "' WHERE CNPJ ='" + CNPJ + "';";

*exec*(sql);

}

#### Leitura de dados

public List<Atributos> read()

{

List<Atributos> atributos = new ArrayList<>();

try {

ResultSet rs = *conectar*().prepareStatement("SELECT \* FROM `atributos`").executeQuery();

while(rs.next()) {

Atributos atributo = new Atributos();

atributo.setNomeFantasia(rs.getString(2));

atributo.setRazaoSocial(rs.getString(3));

atributo.setCNPJ(rs.getString(4));

atributo.setCNAE(rs.getString(5));

atributo.setCNAECodigo(rs.getString(6));

atributo.setCEP(rs.getString(7));

atributo.setAbertura(rs.getString(8));

atributo.setTelefone(rs.getString(9));

atributo.setEmail(rs.getString(10));

atributo.setLogradouro(rs.getString(11));

atributo.setstatus(rs.getString(12));

atributos.add(atributo);

}

} catch (Exception e) {

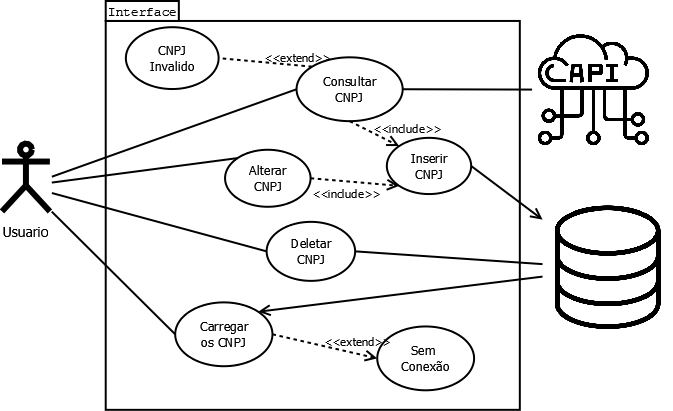
}

return atributos;

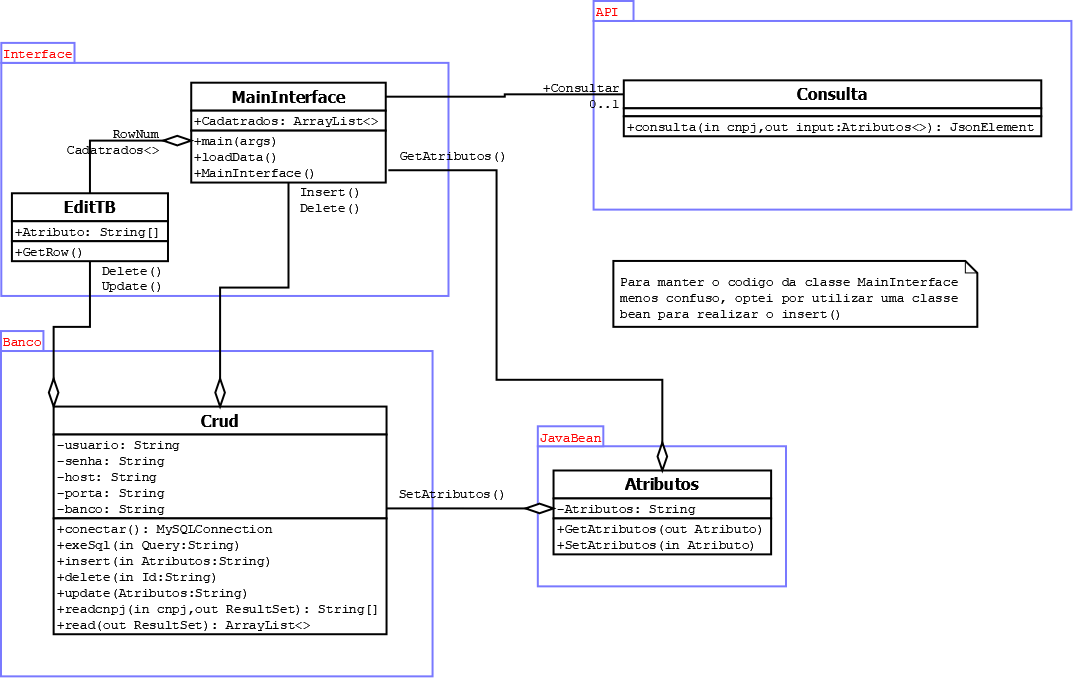
}

## 2.4 DIAGRAMAS

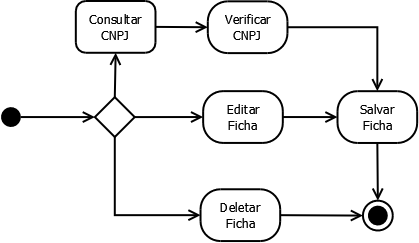
### 2.4.1 CASO DE USO

****

### 2.4.2 CLASSE

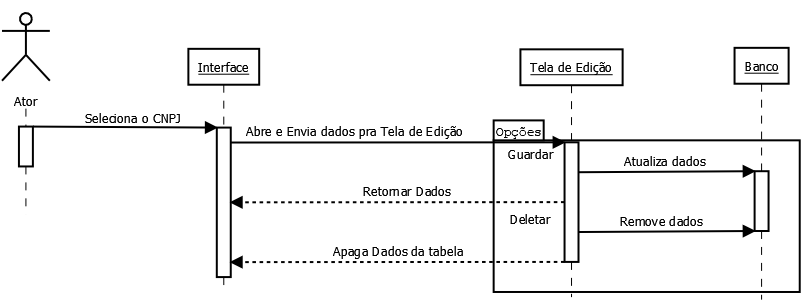


### 2.4.3 ATIVIDADES

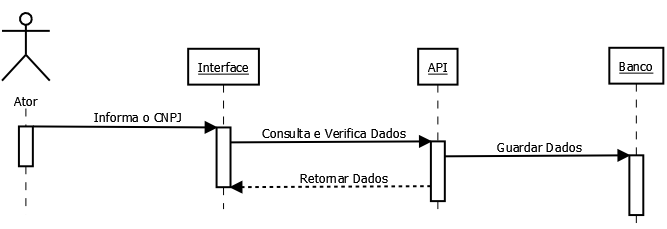


### 2.4.4 SEQUÊNCIA

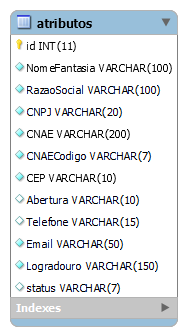
Edição ou Remoção



Consulta



### 2.4.5 MER



## 2.8 REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS

### 2.8.1 REQUISITOS FUNCIONAIS

* **RF1 –** O sistema deve permitir a exclusão de fichas;
* **RF2 –** O sistema deve permitir a edição de fichas;
* **RF3 –** O sistema deve mostrar as fichas cadastradas;
* **RF4 –** O sistema deve armazenar as fichas;
* **RF5 –** O sistema deve realizar a inserção de fichas a partir do CNPJ informado

### 2.8.2 REQUISITOS NÃO FUNCIONAIS

* **RNF1 –** O aplicativo deverá aceitar todos os dados padrões do CNPJ;
* **RNF2 –** A interface terá que ser limpa e fácil de utilizar;
* **RNF3 –** O aplicativo tem como foco o Sistema operacional Windows;
* **RNF4 –** O aplicativo será feito com em JAVA com o banco em SQL;
* **RNF5 –** O Aplicativo terá inserção de dados apenas pela API para evitar dados incorretos.

# 3 CONCLUSÃO

Conforme foi apresentado as problemáticas que foram encontradas, as quais se devem ao fato de haver problemas com o método de guardar os cadastros de clientes e sua vulnerabilidade, foi levantado algumas das necessidades para a melhoria desse processo. Esses requisitos resultaram na criação desse projeto que traz uma solução simples e leve, gerando ao usuário uma maior agilidade, segurança e organização no método de guardar e criar os dados de seus clientes. haja visto que há esta tendencia de informatização dos pequenos negócios o programa pode ajudar essas pequenas empresas acompanharem o mercado, e gerar uma melhor qualidade dos serviços prestados por eles.

# 4 REFERÊNCIAS

Diagramas de caso de uso. 2010. Disponível em: <http://blogdoabu.blogspot.com/2010/02/diagrama-de-casos-de-uso.html>

Informatização dos pequenos negocios, 2022. Disponível em: <https://www.sebrae.com.br/sites/PortalSebrae/ufs/sc/noticias/a-transformacao-digital-esta-presente-nos-pequenos-negocios,fc1603d0db7d4610VgnVCM1000004c00210aRCRD#:~:text=Nos%20%C3%BAltimos%20tr%C3%AAs%20anos%2C%20os,40%25%20possui%20perfil%20no%20Facebook>.

O que é um java bean. Disponível em: <https://pt.stackoverflow.com/questions/22444/o-que-%C3%A9-e-para-que-serve-um-java-bean>