

---

# **Documento de Projeto de Software**

---

**Daniel G. Nascimento**

São Paulo  
2021

# Sumário

1	Descrição Geral do Sistema	2
1.1	Objetivos	2
1.2	Limites e Restrições	2
2	Tecnologias e ferramenta	3
2.1	Prós e contras das plataformas	3
3	Aplicação Alfa	4
3.1	Requisitos do Sistema	4
3.1.1	Requisitos Funcionais	4
3.1.2	Requisitos Não-funcionais	4
3.2	Diagramas do sistema	5
3.2.1	Modelo do Banco de Dados Relacional	5
3.2.2	Diagrama de Classes	5
4	Aplicação Bravo	6
4.1	Requisitos do Sistema	6
4.1.1	Requisitos Funcionais	6
4.1.2	Requisitos Não-funcionais	6
4.2	Diagramas do sistema	7
4.2.1	Modelo do Banco de Dados Relacional	7
4.2.2	Diagrama de Classes	7
5	Aplicação Charlie	8
5.1	Requisitos do sistema	8
5.1.1	Requisitos Funcionais	8
5.1.2	Requisitos Não-Funcionais	8
5.2	Diagramas do sistema	9
5.2.1	Diagrama de classe	9
6	Implementação e descrição do código	10

# **1 Descrição Geral do Sistema**

Nos dias atuais, devido ao grande e constante fluxo de dados, se faz necessário a elaboração de uma solução que ofereça armazenamento, processamento e disponibilização de dados, sempre considerando que tudo deve estar conforme as boas práticas de segurança em TI. Para isso foi idealizado com uma possível solução de armazenamento e consulta de dados.

## **1.1 Objetivos**

Para atender a demanda observada, foi idealizado três aplicações. A primeira delas, que chamamos de Base Alfa, é extremamente sensível e deve ser protegida com os maiores níveis de segurança, em contraponto a isso, ela não precisa ser tão performática. A segunda, é a aplicação Bravo que também possui dados críticos, mas ao contrário da aplicação Alfa, o processamento precisa ser um pouco mais rápido performático. A última é aplicação Charlie, que não possui nenhum tipo de dado crítico, mas precisa ser extremamente performática.

## **1.2 Limites e Restrições**

O sistema planejado contempla a parte de backend. Com isso, gateways, frontend, detalhamento de banco de dados e outros recursos podem ser acoplados e/ou complementados posteriormente.

## 2 Tecnologias e ferramenta

A fim de ter uma maior gama de opções, foi idealizado duas possibilidades de implantação das aplicações, uma pública e outra privada. Ambas suportam as tecnologias idealizadas como solução.

	Plataforma pública	Plataforma privada
Implementação	Java	Java
BD Alfa	RDS/Aurora (AWS)	MySQL
BD Bravo	RDS/Aurora (AWS)	MySQL
BD Charlie	DynamoDB (AWS)	MongoDB
Log	AWS	Elastic (ELK)
Pipeline	AWS / Heroku	Jenkins
Arquitetura	Clean Architecture	Clean Architecture
Comunicação	REST	REST
Repositório de código	GitHub	GitHub

### 2.1 Prós e contras das plataformas

Para elucidar um pouco as diferenças entre as plataformas, são elencados alguns prós e contras de cada plataforma.

#### Prós:

- Plataforma pública: disponibilidade; baixo custo para *PoC* (*proof of concept*); abrangência geográfica com baixa latência; redundância “automática”;
- Plataforma privada: segurança; custos planejados;

#### Contra:

- Plataforma pública: custo elevado para pequenas empresas; dependência de conexão com internet.
- Plataforma privada: possíveis gargalos; orquestração (criação, sustentação, implantação); manutenção (preventiva e corretiva);

## 3 Aplicação Alfa

A aplicação Alfa é idealizada para ser uma aplicação que segue os maiores níveis de segurança que a empresa puder implantar. Devido a esse fato, a performance no tempo de resposta das requisições acaba sendo mais custosa.

### 3.1 Requisitos do Sistema

#### 3.1.1 Requisitos Funcionais

Os requisitos funcionais, são as ações que o usuário pode realizar na aplicação. São eles:

ID	Funcionalidade	Prioridade
RF 01	Permitir a visualização dos dados pessoais do cliente	Importante
RF 02	Permitir o cadastro de novos clientes	Importante
RF 03	Permitir a alteração de dados do cliente	Importante

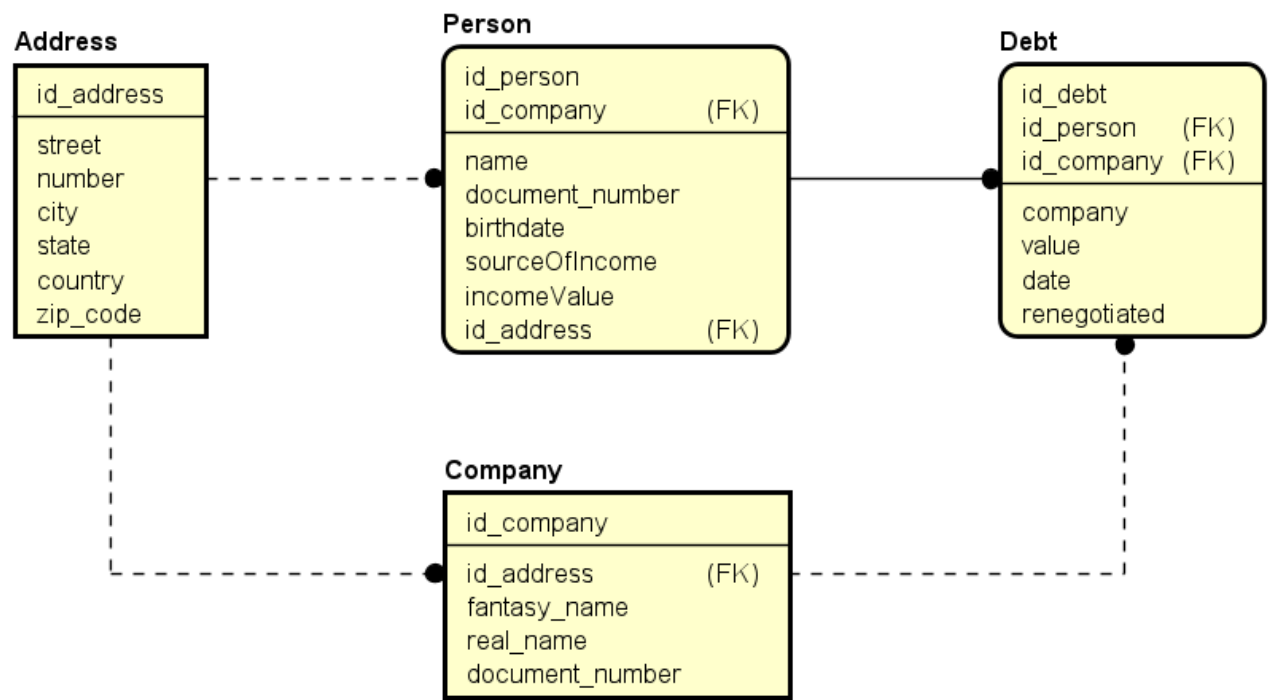
#### 3.1.2 Requisitos Não-funcionais

Para atingir as especificações idealizadas, o sistema deve cumprir os requisitos técnicos que são expostos a seguir.

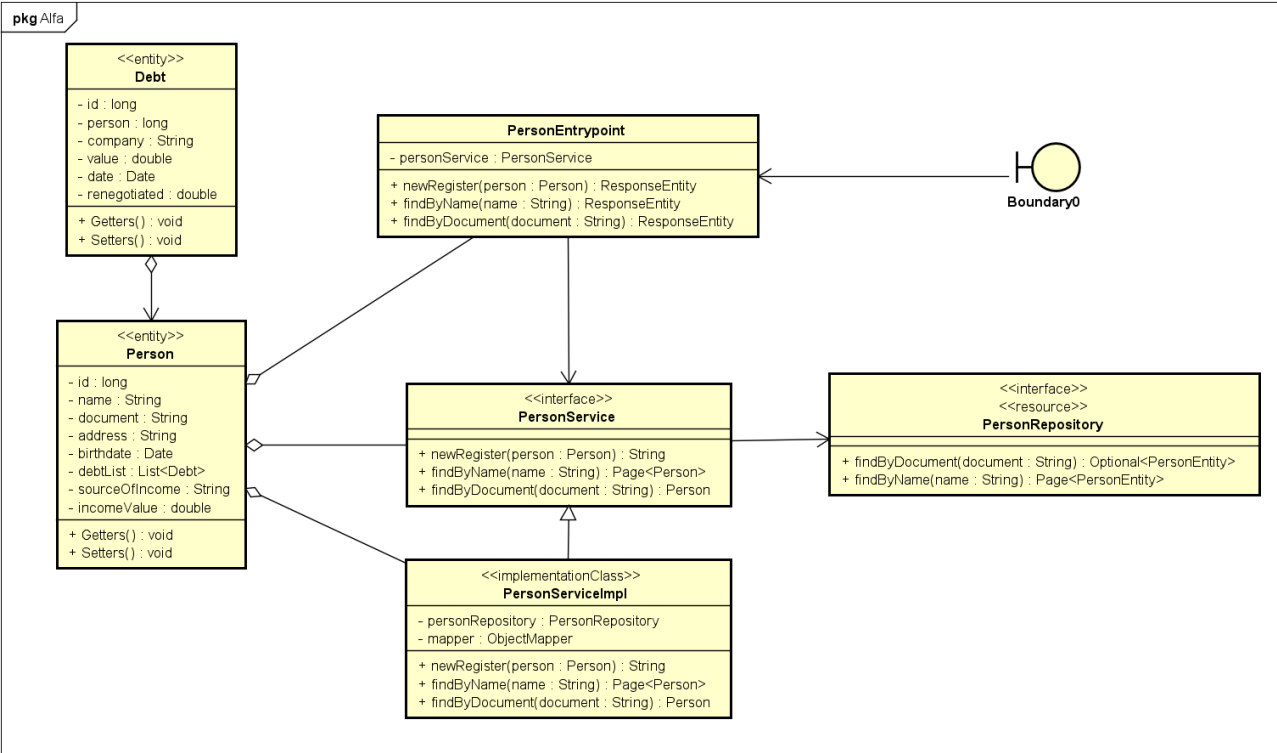
ID	Requisito	Categoria
RNF 01	O sistema deve permitir acesso apenas para requisições que contenham certificado reconhecido e autorizado pelo sistema.	Confiabilidade
RNF 02	O sistema deve permitir acesso apenas para requisições quem contenham <i>token</i> válido ou <i>refresh token</i> valido.	Confiabilidade
RNF 03	O sistema deve criptografar todos os dados trafegados	Segurança
RNF 04	O sistema deve armazenar criptografado todos os dados sensíveis de clientes.	Segurança
RNF 05	O sistema deve ser desenvolvido na linguagem Java	Implementação
RNF 06	O sistema deve se comunicar com um banco de dados relacional	Implementação
RNF 07	O sistema deve armazenar os logs emitidos, para viabilizar auditorias.	Legal

3.2 Diagramas do sistema

3.2.1 Modelo do Banco de Dados Relacional



3.2.2 Diagrama de Classes



## 4 Aplicação Bravo

A aplicação Bravo é idealizada para ser uma aplicação que visa equilibrar segurança e performance.

### 4.1 Requisitos do Sistema

#### 4.1.1 Requisitos Funcionais

Os requisitos funcionais, são as ações que o usuário pode realizar na aplicação. São eles:

ID	Funcionalidade	Prioridade
RF 01	Permitir a visualização dos dados pessoais do cliente.	Importante
RF 02	Permitir a coleta de dados, por usuários com acesso privilegiado, para fins de aprendizado de máquina.	Importante
RF 03	Permitir a consulta de bens do cliente, conforme o número informado do documento pessoal do cliente.	Importante

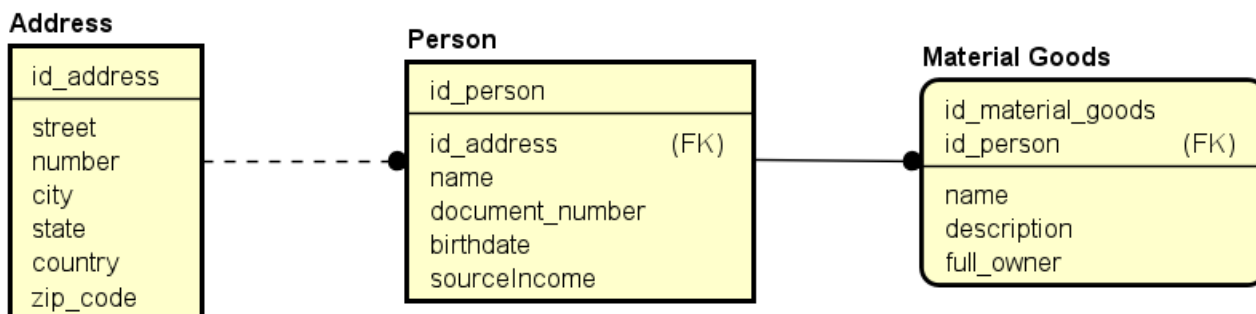
#### 4.1.2 Requisitos Não-funcionais

Para atingir as especificações idealizadas, o sistema deve cumprir os requisitos técnicos que são expostos a seguir.

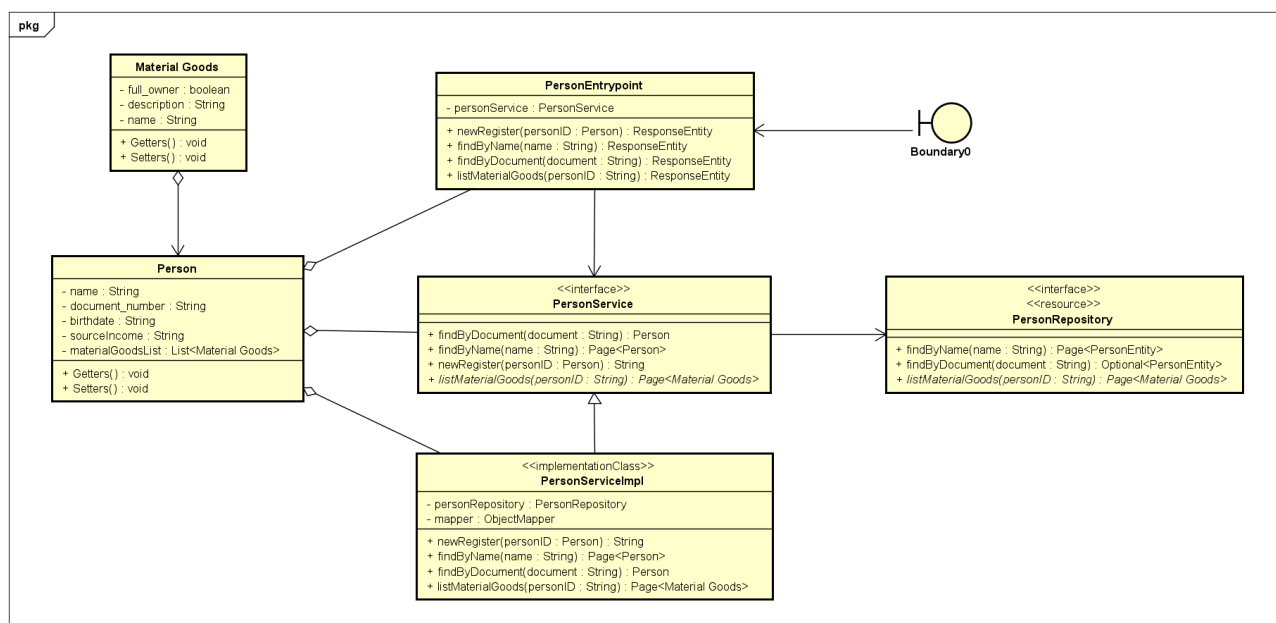
ID	Requisito	Categoria
RNF 01	O sistema deve permitir acesso apenas para requisições quem contenham <i>token</i> válido ou <i>refresh token</i> valido.	Confiabilidade
RNF 02	O sistema deve armazenar criptografado todos os dados sensíveis de clientes.	Segurança
RNF 03	O sistema deve ser desenvolvido na linguagem Java	Implementação
RNF 04	O sistema deve se comunicar com um banco de dados relacional	Implementação
RNF 05	O sistema deve armazenar os logs emitidos, para viabilizar auditorias.	Legal

## 4.2 Diagramas do sistema

### 4.2.1 Modelo do Banco de Dados Relacional



### 4.2.2 Diagrama de Classes





## 5 Aplicação Charlie

A aplicação Charlie é idealizada para ser uma aplicação ágil, performática, pois tem um fluxo intenso de requisições simultaneamente. Suas requisições devem ser respondidas no menor tempo possível, mesmo com grandes volumes de requisições.

### 5.1 Requisitos do sistema

#### 5.1.1 Requisitos Funcionais

Os requisitos funcionais, são as ações que o usuário pode realizar na aplicação. São eles:

ID	Funcionalidade	Prioridade
RF 01	Permitir a consulta de transações do cliente, conforme o número informado do documento pessoal do cliente.	Importante
RF 02	Permitir a consulta de dados relacionados a compras, conforme o número informado do documento pessoal do cliente.	Importante
RF 03	Permitir a consulta sobre a ultima data que o cliente consultou seus dados pessoais.	Importante

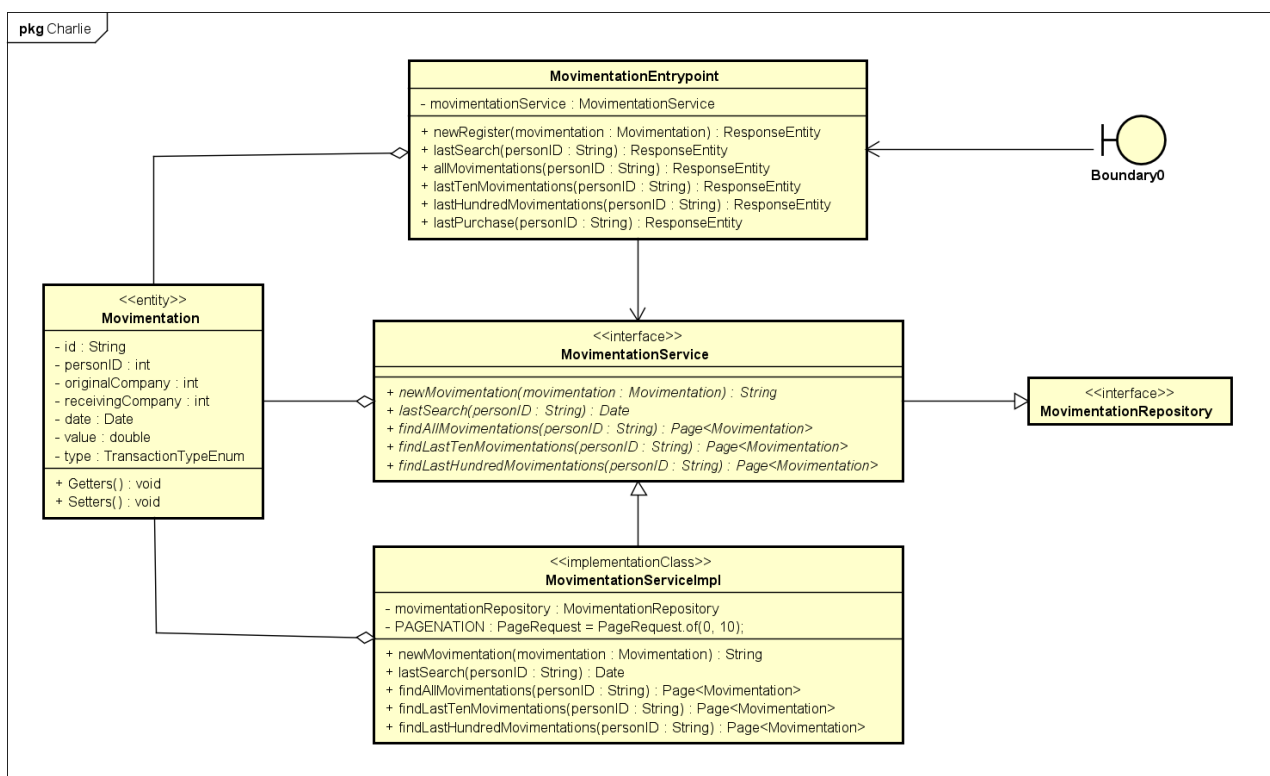
#### 5.1.2 Requisitos Não-Funcionais

Para atingir as especificações idealizadas, o sistema deve cumprir os requisitos técnicos que são expostos a seguir.

ID	Requisito	Categoria
RNF 01	O sistema deve permitir acesso apenas para requisições quem contenham <i>token</i> válido ou <i>refresh token</i> valido.	Confiabilidade
RNF 02	O sistema deve conseguir processar um volume de pelo menos 300 mil requisições por dia.	Desempenho
RNF 03	O sistema deve responder as requisições em no máximo 250 ms.	Desempenho
RNF 04	O sistema deve se comunicar com um banco de dados não relacional	Implementação
RNF 05	O sistema deve armazenar os logs emitidos, para viabilizar auditorias.	Legal

## 5.2 Diagramas do sistema

### 5.2.1 Diagrama de classe



## 6 Implementação e descrição do código

A implementação e descrição do funcionamento do código podem ser encontradas no arquivo 'README' disponível em: <https://github.com/DNascto/projeto>