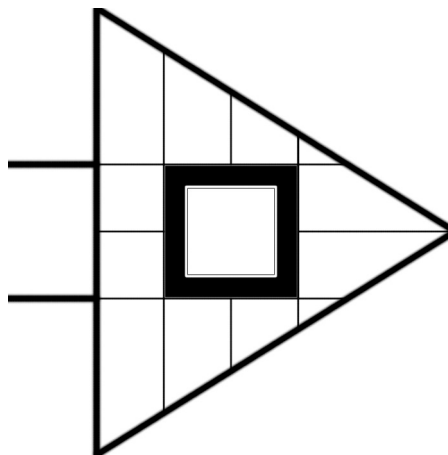




## ConverteLhe



Lucas Kawaji Fukumoto

Orientador Prof. Carlos Aragão

Disciplina Laboratório de Engenharia de Software  
Prof. Carlos Henrique Verissimo

2024

## Sumário

Resumo executivo.....	3
1 Sobre o projeto.....	4
2 Contexto.....	5
3 Necessidades identificadas.....	5
4 Solução.....	5
5 Gestão de projetos.....	6
6 Protótipos.....	8
7 Especificações sobre a implementação do sistema.....	9
8 Requisitos funcionais.....	10
9 Requisitos não funcionais.....	10
10 Diagramação básica.....	11
11 Arquitetura e ecossistema da solução de software.....	15
12 Banco de dados.....	15
Considerações finais.....	16
Referencias.....	17
Anexos e apêndices.....	18

## **Resumo executivo**

Este relatório técnico é referente a proposta técnica de um website e API de composição de processos. São apresentados o propósito do website, sua arquitetura, e o gerenciamento de seu desenvolvimento

## **1 Sobre o projeto**

Este projeto tem como objetivo o desenvolvimento de um website de composição de processos e, usando essas composições, processamento de dados. Será utilizada a metodologia SCRUM, composta de 5 sprints, cada uma com a duração de 21 dias, no período do segundo semestre de 2024; e será desenvolvido inteiramente por Lucas Kawaji Fukumoto e orientado pelo professor Carlos Alberto Pina Aragão.

## 2 Contexto

Todo usuário de computador já passou pela situação de precisar converter algum valor em algum outro valor, por exemplo converter binário para decimal, aplicar uma função matemática, substituir Expressões Regulares, colocar valores num texto, e etc.

Para resolver esse problema existem diversos programas e websites que convertem as bases, resolvem os cálculos, substituem no texto, formatam os valores. Porém se for necessário fazer mais de uma dessas operações é necessário utilizar múltiplos desses websites, e pior, se for necessário realizar essas operações para uma lista de valores, o tempo necessário para essa conversão aumenta drasticamente, causando tédio e monotonia; que, segundo Loukidou (2009), é prejudicial à performance, atenção e saúde da pessoa, e por conta da redução da atenção pode levá-la a cometer erros.

## 3 Necessidades identificadas

Os usuários precisam realizar diversos tipos de processamentos numa lista de dados armazenados num arquivo de texto, porém não possuem conhecimento sobre expressões regulares ou programação; ou, embora tenham conhecimento de programação, não querem perder tempo criando um programa para tal processamento, pois a frequência que eles tem que realizar esse processamento é muito baixa.

## 4 Solução

O website ConverteLhe permite e facilita a tradução da necessidade de processamento do usuário para uma composição de processamentos, e a utilização dessa composição para o processar uma lista de dados.

## 5 Gestão de projetos

Organização da equipe:

– Lucas Kawaji Fukumoto: tudo

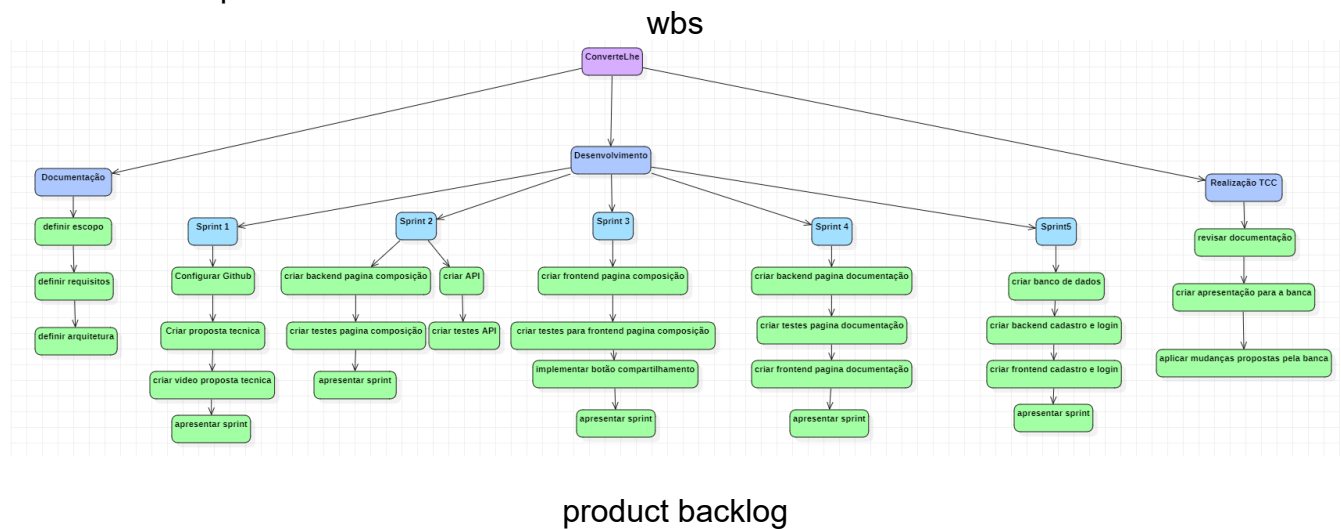
Organização das atividades:

– Lucas Kawaji Fukumoto: tudo

Organização da comunicação:

– Microsoft Teams: tudo

Gestão de tempo:



ID da História/caso de uso	História/Caso de Uso	Crterios de Aceitação	Prioridade	Sprint	Status
HU001	Como um usuário eu gostaria de conseguir montar processamentos personalizados para as minhas necessidades específicas	>>Eu quero ter uma interface one eu possa ver quais processos eu consigo usar e quero conseguir inseri-los no meu processamento	1	3	Não iniciada
HU002	Como um usuário eu gostaria de conseguir utilizar o processamento que eu montei para processar meus dados	>>Eu quero conseguir mandar uma string ou um arquivo para ser processado e ser retornado os dados processados no mesmo formato que eu enviei >>Eu quero que o sistema me fale caso meu processamento tenha erros	1	2	Não iniciada
HU003	Como usuário eu quero conseguir compartilhar os processamentos que eu criei com outras pessoas	>>Eu quero conseguir criar um endereço que direcione ao processamento que eu criei	2	3	Não iniciada
HU004	Como usuário eu gostaria de conseguir utilizar a API do sistema no meu projeto	>>Eu quero conseguir fazer requisições https para o API do sistema	3	5	Não iniciada
HU005	Como usuário eu gostaria que existisse uma pagina de documentação para que eu consiga saber sobre os detalhes sobre cada processo	>>Eu quero uma pagina para cada processo	2	4	Não iniciada

## sprint backlog

SPRINT 1 BACKLOG				
ID	Caso de uso	Tarefa	Prioridade	Recurso/ Responsável
		configurar github	1	LucasF
		criar proposta tecnica	1	LucasF
		criar video proposta tecnica	1	LucasF
		apresentar sprint	1	LucasF
SPRINT 2 BACKLOG				
ID	Caso de uso	Tarefa	Prioridade	Recurso/ Responsável
	NCU02 Processar Dados	backend composição	1	LucasF
		teste backend composição	2	LucasF
		API	1	LucasF
		teste API	2	LucasF
		apresentar sprint	1	LucasF
SPRINT 3 BACKLOG				
ID	Caso de uso	Tarefa	Prioridade	Recurso/ Responsável
	NCU01 Composição de Processos	criar frontend composição	1	LucasF
		testes frontend composição	2	LucasF
	NCU03 Url de compartilhamento	implementar botão comaprtilhamento	2	LucasF
		apresentar sprint	1	LucasF
SPRINT 4 BACKLOG				
ID	Caso de uso	Tarefa	Prioridade	Recurso/ Responsável
	HU005	criar backend pagina documentação	1	LucasF
		criar testes pagina documentação	2	LucasF
		criar frontend pagina documentação	1	LucasF
		apresentar sprint	1	LucasF
SPRINT 5 BACKLOG				
ID	Caso de uso	Tarefa	Prioridade	Recurso/ Responsável
	NCU04 Manter usuário	cadastro/login frontend	2	LucasF
		login	1	LucasF
		cadastro	1	LucasF
		mudar senha	1	LucasF
		apresentar sprint	1	LucasF

## 6 Protótipos

Interface de composição



A interface de composição é a tela principal e mais importante do projeto, ela é dividida em 3 partes: composição, dados, e componentes. Nela o usuário poderá compor o seu processamento na parte de composição (metade de cima); utilizando os processamentos da parte de componentes (parte inferior direita) e processar seus dados na parte de dados (parte inferior esquerda).

Na parte de dados existem 2 ícones, quando clicados, um deles leva à interface de usuário e outro gera o url que leva ao processamento atual.

Interface de usuário

O diagrama representa a interface de usuário. A parte superior é uma caixa cinza com campos de login. Há um rótulo 'email:' seguido de um campo de texto branco. Abaixo dele, um rótulo 'password:' seguido de um campo de texto branco. À direita do campo de senha há um link azul 'forgot password'. Abaixo dos campos há dois botões cinza: 'register' e 'login'. A parte inferior é outra caixa cinza com informações de status. Ela contém o texto 'email: blablabla@email.com', 'API key: \*\*\*\*\*' e 'status: limit exceded' em vermelho.



A interface de usuário é responsável pelo gerenciamento de tudo relacionado ao usuário: cadastro, login, recuperação de senha, e verificação de API key.

## 7 Especificações sobre a implementação do sistema

Gerenciamento do projeto:

LibreOffice Calc → product backlog e sprint backlog

Github → controle de versão

StarUML → diagrama de classe, sequencia, caso de uso, estado, atividade, implantação e wbs

Microsoft Teams → comunicação

Prototipação:

Figma → protótipo de tela

Frameworks:

OTP → bibliotecas, e ferramentas para o uso do Erlang

Linguagens de programação:

Erlang → servidor

JavaScript → frontend

Banco de dados:

Supabase

## 8 Requisitos funcionais

RF001 – Processar dados – essencial

capacidade do usuário utilizar uma composição de processamentos para processar dados

RF002 – Compor processamentos – essencial

capacidade de combinar processamentos para criar uma composição de processamentos

RF003 – Criar url de compartilhamento – essencial

capacidade de gerar uma url correspondente a um processamento específico

RF004 – Manter usuário – importante

capacidade do usuário criar e manter uma conta

## 9 Requisitos não funcionais

RNF001 – Segurança – essencial

Os dados do usuário devem ser criptografados quando trafegando pela internet

RNF002 – Documentação online – importante

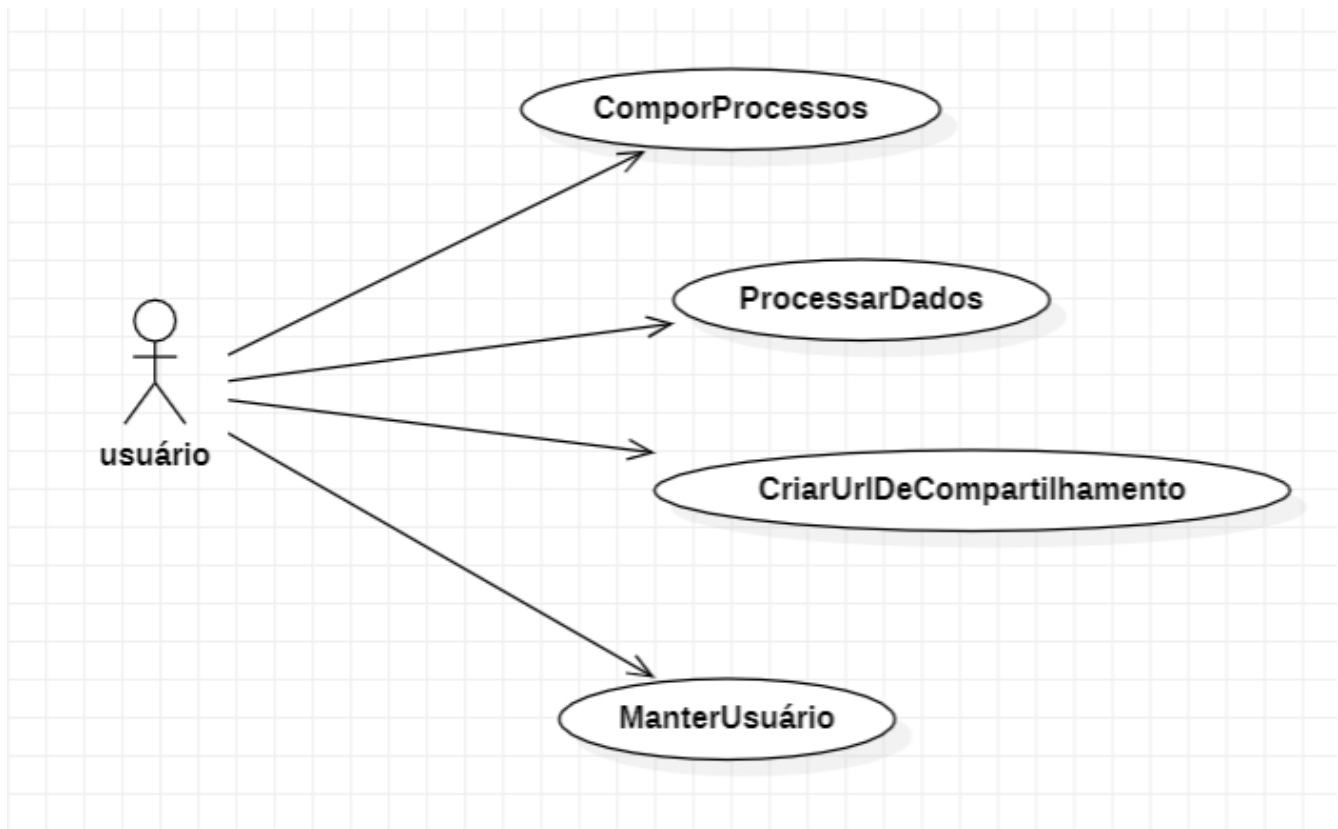
Deve existir uma documentação online para tirar dúvidas dos usuários sobre o funcionamento do sistema

RNF003 – Design pattern – importante

O projeto deve utilizar a arquitetura MVC para melhorar a sua manutenibilidade e qualidade

## 10 Diagramação básica

Diagrama de caso de uso



Atores:

Usuário → refere-se a todo e qualquer visitante do website

CSU001 – Processar dados	
Sumário:	O usuário usa o sistema para processar dados
Ator Primário:	usuário
Ator Secundário: nenhum	
Casos de Uso Associados: CSU001	
<b>Fluxo Principal</b>	
1– O usuário insere ou não a sua API key	
2– O sistema verifica se a composição atual é válida; se não for ir para o fluxo de exceção “Composição invalida”	
3– O sistema verifica se a API key é válida, caso o usuário não tenha mandado uma API key seguir para o	

passo 4, caso a API key seja invalida ir para o fluxo de exceção “API key invalida”
<b>4–</b> O sistema verifica se o usuário tem limite ( utiliza o endereço de ip caso não tenha sido inserido uma API key no passo 1), caso não tenha limite ir para o fluxo de exceção “Limite excedido”
<b>5–</b> O sistema realiza o processamento e retorna o resultado para o usuário
<b>6–</b> O caso de uso termina
<b>Fluxo de Exceção – Composição invalida:</b> <ul style="list-style-type: none"> <li>- O sistema exibe onde e porque a composição é invalida</li> <li>- O caso de uso retorna ao passo 6</li> </ul> <b>Fluxo de Exceção – API key invalida:</b> <ul style="list-style-type: none"> <li>- O sistema exibe que a API key utilizada é invalida</li> <li>- O caso de uso retorna ao passo 6</li> </ul> <b>Fluxo de Exceção – Limite excedido:</b> <ul style="list-style-type: none"> <li>- O sistema exibe que o limite de processamentos foi excedido</li> <li>- O caso de uso retorna ao passo 6</li> </ul>
<b>Pós-condições:</b> <ul style="list-style-type: none"> <li>a. o limite do usuário é incrementado</li> </ul>
<b>Requisitos: RF001</b>
<b>Regras de Negócio: RN001, RN002</b>

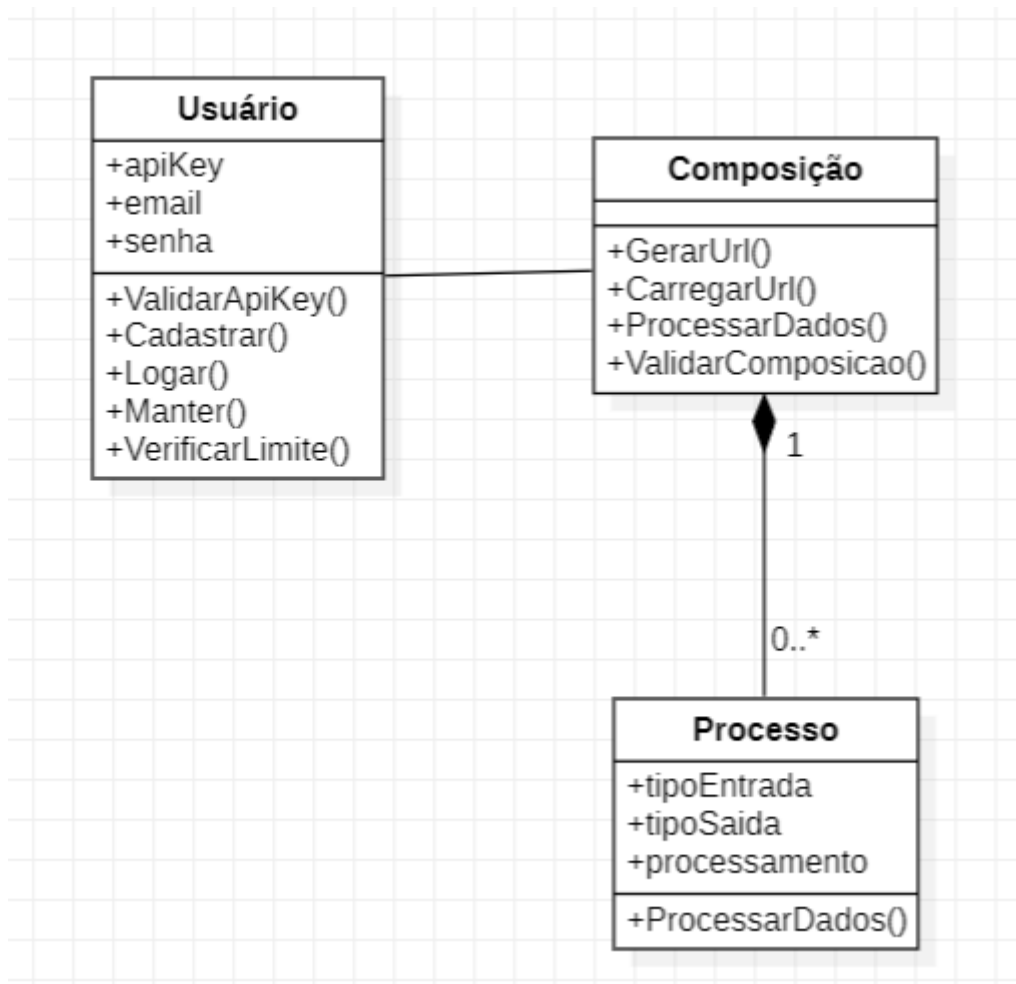
<b>CSU002 – Compôr Processos</b>	
<u>Sumário:</u>	O usuário usa o sistema criar uma sequência de processamentos
Ator Primário:	Usuário
Ator Secundário: não se aplica	
Casos de Uso Associados:	
<b>Pré-requisitos:</b> Há uma composição na página (vazia ou não)	
<b>Fluxo Principal</b> <p><b>1–</b> O usuário escolhe o processamento, insere os parâmetros e a posição do processamento</p> <p><b>2–</b> O sistema verifica se há erros na composição, se houver ir exibe qual é, e onde está o erro</p> <p><b>3–</b> O caso de uso termina</p>	
<b>Requisitos: RF002</b>	
<b>Regras de Negócio: RN001</b>	

<b>CSU003 – Criar URL de compartilhamento</b>	
<u>Sumário:</u>	O usuário usa o sistema criar um URL referente a uma sequência de pro-

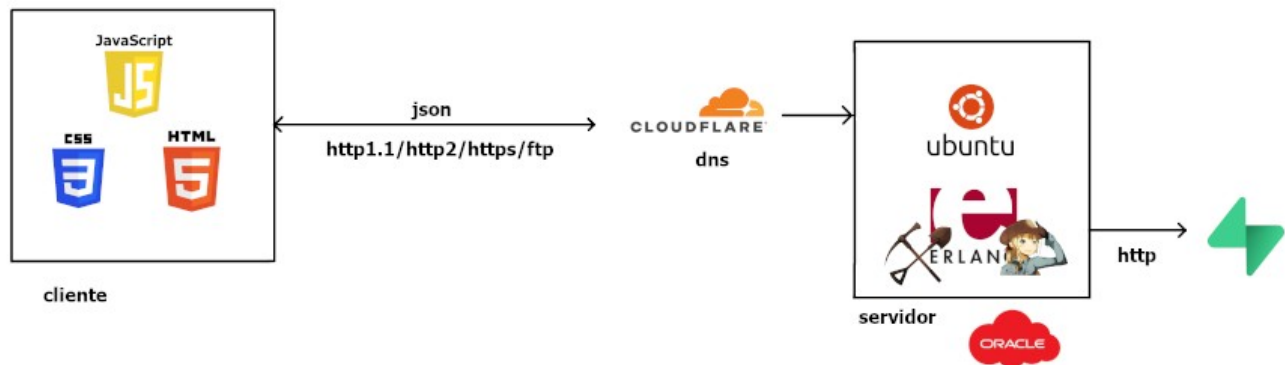
	cessamentos
Ator Primário:	Usuário
Ator Secundário:	não se aplica
Casos de Uso Associados:	
<b>Pré-requisitos:</b>	Há uma composição na página (vazia ou não)
<b>Fluxo Principal</b>	<p>1– O usuário clica no botão de gerar URL</p> <p>2– O sistema converte a Composição atual (independentemente de erros de composição) num URL</p> <p>3– O caso de uso termina</p>
<b>Requisitos:</b>	RF003
<b>Regras de Negócio:</b>	Não se aplica

<b>CSU004 – Manter usuário</b>	
<u>Sumário:</u>	O usuário usa o sistema para criar ou manter uma conta
Ator Primário:	Usuário
Ator Secundário:	não se aplica
Casos de Uso Associados:	
<b>Pré-requisitos:</b>	
<b>Fluxo Principal</b>	<p>1– O usuário clica no ícone de usuário</p> <p>2– O sistema exibe a página do usuário</p> <p>3– O usuário insere o seu e-mail e senha nos respectivos campos</p> <p>4– Caso o usuário clique em “registrar” ou “esqueci minha senha” ir para o fluxo de alternativo “Mudar senha”</p> <p>5– O usuário clica em login</p> <p>6– O sistema verifica se a combinação de e-mail e senha existe, se não existir o sistema exibe “senha invalida”, caso contrário ele exibe as informações do usuário: e-mail, API key, e limite</p> <p>7– O caso de uso termina</p>
<b>Fluxo alternativo – Mudar senha:</b>	<ul style="list-style-type: none"> <li>- O sistema manda um e-mail para o e-mail inserido contendo um link para a confirmação da mudança de senha</li> <li>- O usuário clica no link e sua senha será alterada para a senha inserida</li> <li>- O caso de uso retorna ao passo 7</li> </ul>
<b>Requisitos:</b>	RF004
<b>Regras de Negócio:</b>	Não se aplica

## Diagrama de classes



## 11 Arquitetura e ecossistema da solução de software



Será utilizado o oracle cloud para criar uma máquina virtual ubuntu 22.04, que executará o projeto Erlang.mk no background. Para abrir as portas 80 (http) 443(https) e 20 (ftp) será utilizada a biblioteca Cowboy; para o protocolo dns será utilizado o Cloudflare; e a conexão com o supabase será por meio de requisições http.

O ambiente de desenvolvimento será um wsl (Windows Subsystem for Linux) ubuntu, rodando o Erlang OTP 24; e a implantação será feita manualmente utilizando FilleZilla e Putty, e já que o Erlang.mk suporta hot code loading não haverá downtime.

## 12 Banco de dados

Usuário			
		Tamanho	Constraint
id	int		PK
email	varchar	40	unique not null
senha	varchar	30	not null
API key	varchar	20	unique not null
limite	int		not null

## Considerações finais

Embora o Erlang seja conhecido por ter grande escalabilidade por meio de clusters, esse projeto utilizará apenas uma máquina virtual, pois é o máximo que foi possível obter gratuitamente.

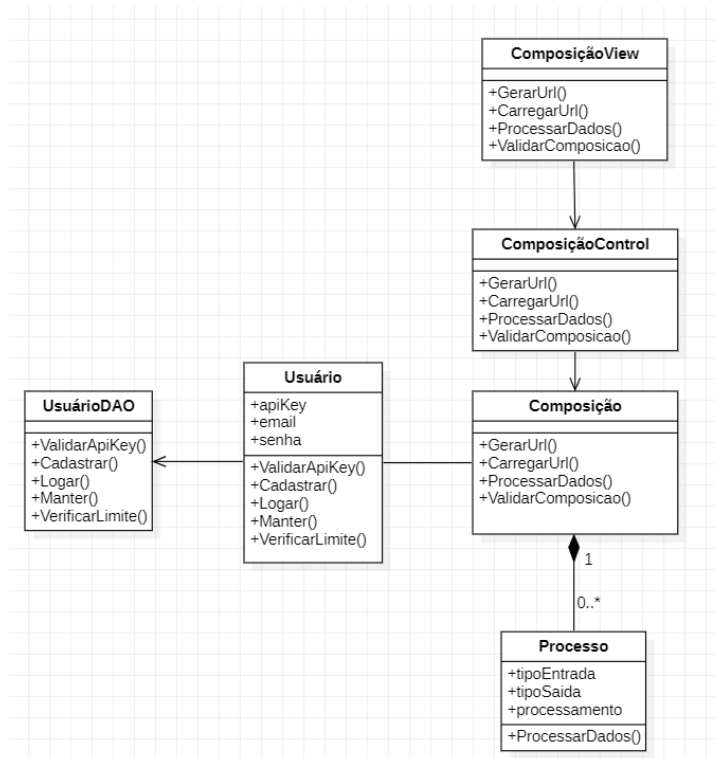


## **Referencias**

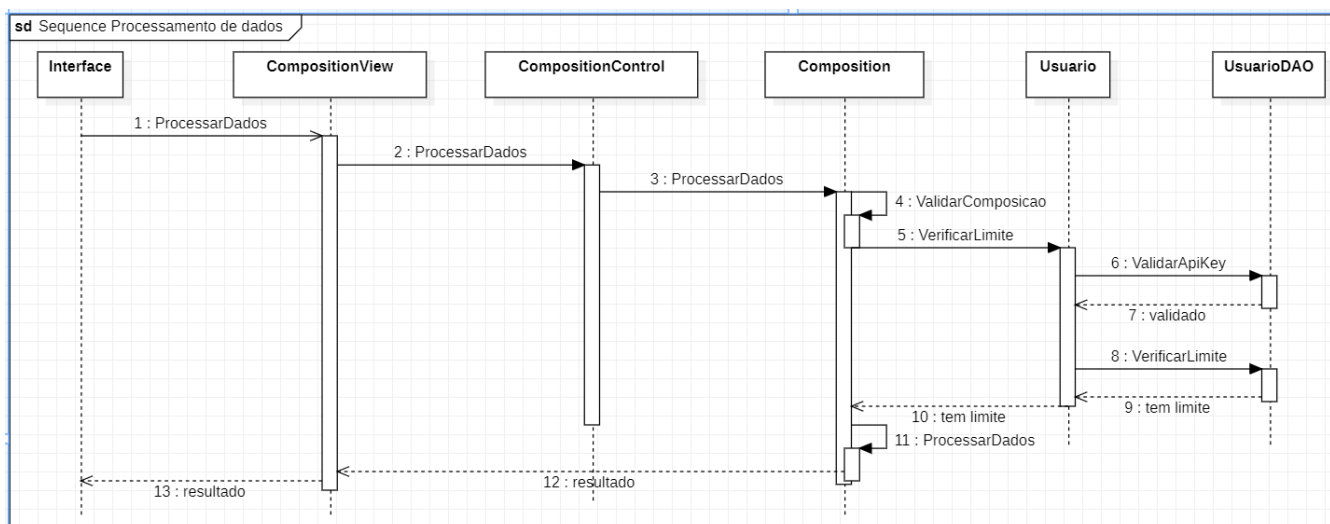
LOUKIDOU, Lia; LOAN-CLARKE, John; DANIELS, Kevin. Boredom in the workplace: More than monotonous tasks. *International Journal of Management Reviews*, v. 11, n. 4, p. 381-405, 26 out. 2009.

## Anexos e apêndices

### Apendice001 – arquitetura MCV na requisição do processamento



### Apendice002 – diagrama de sequencia do processamento



## Apendice003 – diagrama de atividade do processamento

