

## **Sistema Rede Lógica (SRL)**

## PREFÁCIO

Versão	Data	Autor	Razão
1.0.0	13/08/2019	Cleiton Modenes	Criação do documento
1.0.1	18/08/2019	Lucas Ribeiro	Edição do documento
1.0.2	29/08/2019	Vinicius Cavalheiro	Inserido diagrama de sequência
1.0.3	29/08/2019	Alan Correia Landi	Inserido diagrama de Sequência Re13

## GLOSSÁRIO

Controle (Controller): É a camada responsável entre a interface e o SGBD.

Criptografia: É um mecanismos que garantem a segurança da informação na comunicação dos dados, desta forma garantindo a proteção dos dados durante a comunicação para que evite a quebra da privacidade dos dados.

HTML 5: (Hypertext Markup Language, versão 5) é uma linguagem de marcação para a World Wide Web e é uma tecnologia chave da Internet.

Laravel Framework: é um Framework PHP utilizado para o desenvolvimento web, que utiliza a arquitetura “Model View Controller” e tem como principal característica ajudar a desenvolver aplicações seguras e performáticas de forma rápida, com código limpo e simples, já que ele incentiva o uso de boas práticas de programação e utiliza o padrão PSR-2 como guia para estilo de escrita do código.

Login; Logout: Login: Login é definido basicamente como a forma de acesso ao sistema e Logout sendo seu oposto ou seja a forma de sair do sistema.

Model: Camada utilizada como parâmetro e retorno entre as chamadas das camadas View e Controller.

MVC: O MVC é um padrão de arquitetura de desenvolvimento de software. Dividido entre camadas, sendo elas modelo, visão e controle.

MySQL Server: Sistema gerenciador de banco de dados desenvolvido pela Oracle

*On-line; Off-line*: *On-line* é um termo utilizado para descrever a condição de estar conectado a algo, podendo ser à um computador ou uma rede, podendo ela ser ou não a internet. *Off-line* é o oposto de on-line, assim sendo sua condição de não conectado à um computador ou rede.

PHP: (Personal Home Page) é uma linguagem interpretada livre, usada originalmente apenas para o desenvolvimento de aplicações presentes e atuantes no lado do servidor, capazes de gerar conteúdo dinâmico na World Wide Web.

SGBD: É uma sigla que abrevia Sistema Gerenciador de Banco de dados, coleção de softwares responsável pelo armazenamento e manutenção de informações persistentes do computador.

View: Camada composta pelas interfaces visuais do sistema, do qual o usuário realiza toda a interação.

## SUMÁRIO

INTRODUÇÃO .....	5
ARQUITETURA DO SISTEMA .....	6
REQUISITOS NÃO FUNCIONAIS DE USUÁRIO .....	10
REQUISITOS FUNCIONAIS DE USUÁRIO .....	12
REQUISITOS NÃO FUNCIONAIS DE SISTEMA .....	14
REQUISITOS FUNCIONAIS DE SISTEMA .....	15
MODELO DO SISTEMA .....	20
CRONOGRAMA DE EXECUÇÃO .....	23
PLANEJAMENTO DE TESTES E MANUTENÇÃO .....	24

## INTRODUÇÃO

O Sistema Rede Lógica (SRL) é um software desenvolvido para suprir a necessidade de um facilitador organizacional para informações na área de redes.

Atualmente apesar da grande gama de softwares que são desenvolvidos dia após dia, um sistema de gestão que computa dados de uma ou mais redes é algo em carência no mercado, visando essa necessidade e sempre ponderando quanto à indispensabilidade de um sistema com esse intuito foi nos proposto a elaboração deste projeto para uma melhor Administração de redes do IFSP (Instituto Federal de Educação, Ciência e Tecnologia de São Paulo).

Nosso objetivo é a criação de um software que seja capaz de agrupar informações pertinentes aos administradores e coordenadores de redes, com esse intento a possibilidade de manter informações no sistema para que o usuário possa manter sua(s) rede(s) sempre atualizada evitando duplicação ou erros que acaba contribuindo para má gestão e ou perda de tempo tendo a necessidade de anotações manuais quanto a sua rede. O SRL agrega informações referente a implantação de redes de computadores, organização de Ip's válidos e Vlan em Switch, apresentando a conexão de equipamentos, armazenando log das operações e contando com um sistema de auditoria, separando todas as ferramentas por nível de acesso do usuário, para uma melhor organização e gestão da rede.

Resumo do Sistema

Nome: Sistema Rede Lógica (SRL)

Tipo: Aplicação *Web*

Linguagem: PHP

## ARQUITETURA DO SISTEMA

O sistema desenvolvido utiliza o padrão *Model View Controller* (MVC) para organizar suas estruturas internas. São utilizadas três camadas:

- Aplicação (*View*)

Camada responsável pela interface com o usuário. Desenvolvida em HTML5.

- Controle (*Controller*)

Camada responsável pela comunicação entre a camada de aplicação e o banco de dados. Nessa camada são implementadas as regras de negócio do sistema, bem como as validações dos dados. Desenvolvida na linguagem PHP.

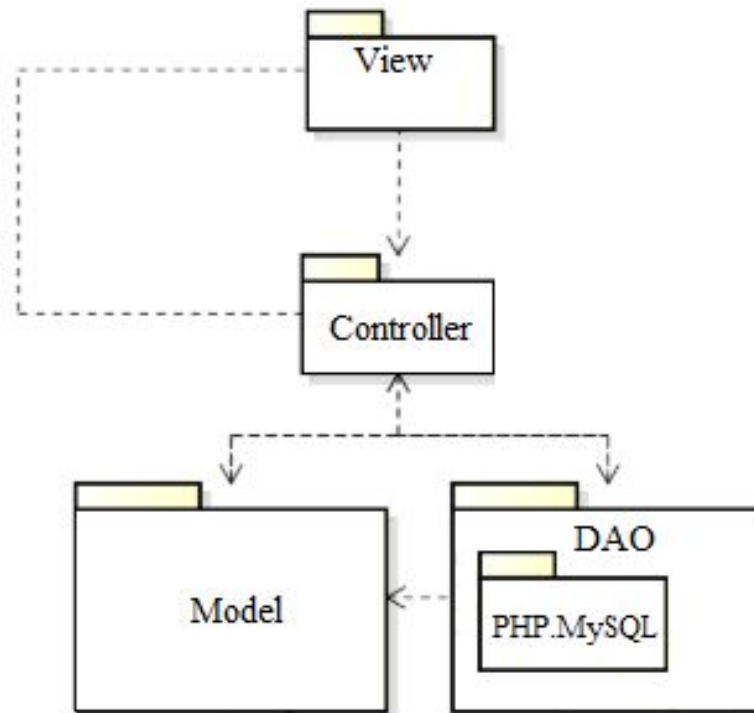
- Modelo (*Model*);

Camada que possui classes representando as tabelas existentes no banco de dados.

Desenvolvida na linguagem PHP.

A Figura 1 exibe um exemplo de comunicação entre as camadas. A camada View, responsável pela interface gráfica, não se comunica diretamente com o banco de dados. Para indicar uma operação no banco de dados, ela faz uma chamada à camada Controle, que faz todas as verificações/operações de regras de negócio, e só então realizará acesso ao banco de dados. Para facilitar a comunicação entre as camadas, objetos da camada Modelo, que representam as tabelas existentes no banco de dados, são utilizadas para transmitir informações entre as camadas.

Figura 1 - Comunicação entre as camadas do *software*



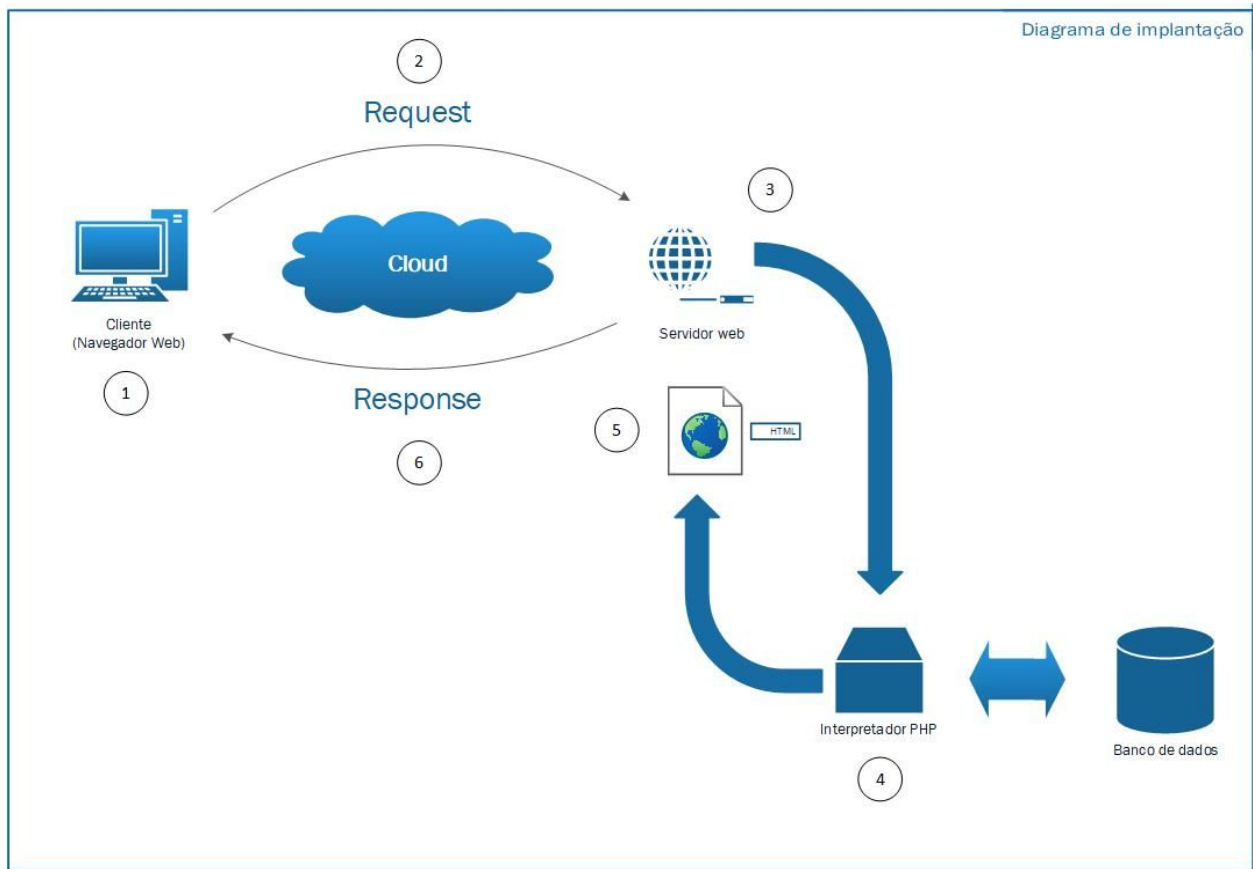
Fonte: próprio autor

O sistema executa em um ambiente web, onde em um servidor será implantada a parte lógica do sistema(PHP.MySQL)

A Figura 2 exibe o diagrama que representa de forma simplificada um esquema de funcionamento de uma aplicação web PHP.



Figura 2 – Diagrama de implantação



Fonte: Próprio autor

No nó 1 tem-se o CLIENTE, como qualquer navegador web. Um navegador web funcionará como interface de exibição das páginas HTML processadas pelo servidor Web.

O nó 2 é a operação de REQUEST. Um request (ou requisição) é a solicitação enviada de um cliente (navegador web) a um servidor web. Esta requisição é disparada por um evento no lado cliente e sempre aguardará por uma posterior resposta (response) do servidor web acionado.

O nó 3 é o SERVIDOR WEB (web server). Ele é responsável por hospedar as páginas web e processar as requisições recebidas, devolvendo-as ao cliente por meio das respostas (response). Em teoria, um servidor web entende apenas HTML. Qualquer requisição que não seja HTML será redirecionada ao INTERPRETADOR (nó 4), que por sua vez transformará todo output em HTML, devolvendo o fluxo ao Servidor Web.

O INTERPRETADOR (nó 4) processa códigos em linguagem de programação, transformando a saída (output) em HTML. O dispositivo de saída padrão do interpretador PHP é o Servidor Web. Em outras palavras, todo código HTML gerado pelo PHP (nó 5) será enviado ao Servidor Web como resposta a requisição do cliente.

Quando o Servidor Web recebe todo o fluxo HTML gerado pelo interpretador PHP (nó 5), este responde ao cliente, sincronizando a requisição recebida (nó 6).

Dessa forma, para cada requisição (REQUEST) haverá uma resposta (RESPONSE) correspondente. Esse par de REQUEST/RESPONSE é o responsável pelo sincronismo da aplicação web.

## **REQUISITOS NÃO FUNCIONAIS DE USUÁRIO**

Descrevem as características gerais do sistema de forma sucinta e de simples entendimento.

### **RNF01 – Utilização**

O sistema sendo uma aplicação web, será acessado a partir de qualquer terminal (celular, tablet, notebook, computador) com acesso a internet.

### **RNF02 – Segurança das informações**

As informações devem ser armazenadas de forma segura.

### **RNF03 - Suporte**

O sistema oferecerá suporte a todos os navegadores que possuam suporte para HTML 5.

### **RNF04 – Disponibilidade**

O usuário só pode acessar o sistema de maneira on-line.

### **RNF05 - Desempenho**

O usuário tem tempo limite para a recuperação de senha quando solicitada.

### **RNF06 - Autenticação**

O usuário para acesso ao sistema, precisa de um email e senhas cadastrados.

### **RNF07 - SGBD**

O Sistema Gerenciador Banco de Dados só pode ser acessado de maneira on-line pelo usuário.

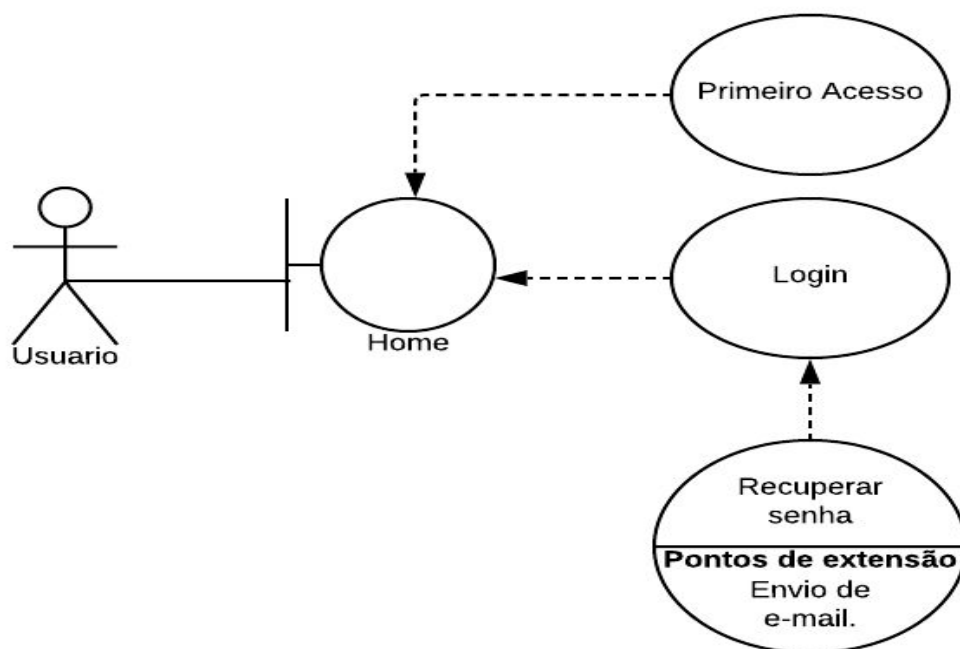
## RNF08 - Usabilidade

O usuário poderá acessar o sistema de maneira on-line, em navegadores, que consigam interpretar HTML 5.

## REQUISITOS FUNCIONAIS DE USUÁRIO

Os requisitos funcionais de usuário descrevem de maneira sucinta e resumida as funcionalidades do sistema. Na figura 3 é demonstrado o diagrama de caso de usos, do qual indica as possibilidades de utilização do sistema por intermédio do usuário

Figura 3 – Diagrama de casos de uso



Fonte: próprio autor

Até o presente momento o único ator do sistema é o usuário(administrador) que é responsável por todas as operações realizadas no sistema. abaixo abordaremos sobre os requisitos funcionais de usuário.

### Requisito RF11 – Primeiro acesso

Em caso do primeiro acesso o usuário ao sistema deverá cadastrar um administrador no sistema.

#### Requisito RF12 - Login

O sistema deve permitir ao usuário devidamente cadastrado realizar login no sistema para conseguir acesso às ferramentas do sistema.

#### Requisito RF13 – Recuperar senha

O Sistema possibilita a recuperação de senha pelo usuário, ao requisitar a recuperação de senha é enviado um link para o email cadastrado pelo usuário do sistema um link para possibilidade de troca de senha, tem durabilidade de 60 minuto até a expiração.

## REQUISITOS NÃO FUNCIONAIS DE SISTEMA

Descrevem as propriedades gerais do sistema de forma completa, indicando características e métricas utilizadas, quando necessário.

### RNF01 - Compatibilidade

O sistema é uma aplicação web, desenvolvido em PHP, sendo o HTML utilizado como linguagem de marcação, devido a esses aspectos e o fato de ser responsivo, o sistema executa em navegadores disponíveis em diversas plataformas desde que a mesma possua suporte para HTML 5 e a plataforma da qual o sistema esteja sendo utilizado possua conexão ativa com a internet.

### RNF02 – Autenticação

Autenticação do usuário é realizada através de email e senha cadastrados, o armazenamento da senha é realizado no Sistema de Gerenciamento de Banco de Dados (SGBD), o armazenamento da senha é feito através de uma hashing (bcrypt) que garante a segurança da senha do usuário.

### RNF03 - Segurança das informações

O sistema utiliza um Sistema Gerenciador de Banco de dados(SGBD) com intuito de armazenar os dados de forma segura. O Sistema Gerenciador de Banco de dados utilizado é o MySQL (phpMyAdmin). O banco de dados será inicialmente armazenado em um servidor web local, do qual o usuário terá que ter acesso de maneira *online*.

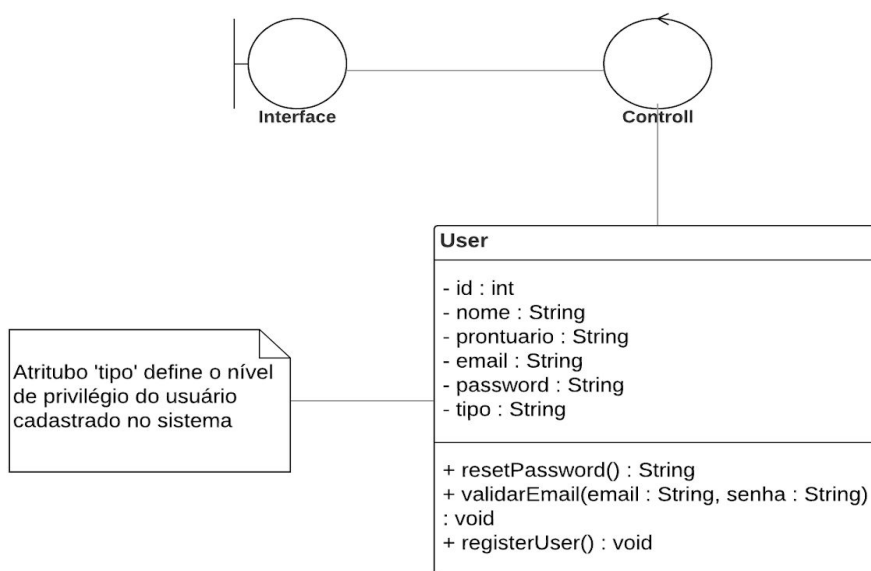
## REQUISITOS FUNCIONAIS DE SISTEMA

Os requisitos funcionais do sistema descrevem as funcionalidades do sistema, de maneira completa, tendo como referência o desenvolvimento. Na Figura 4 é demonstrado o diagrama de classes do sistema.

Figura 4 – Diagrama de classe

### DIAGRAMA DE CLASSE

Lucas Ribeiro | September 13, 2019



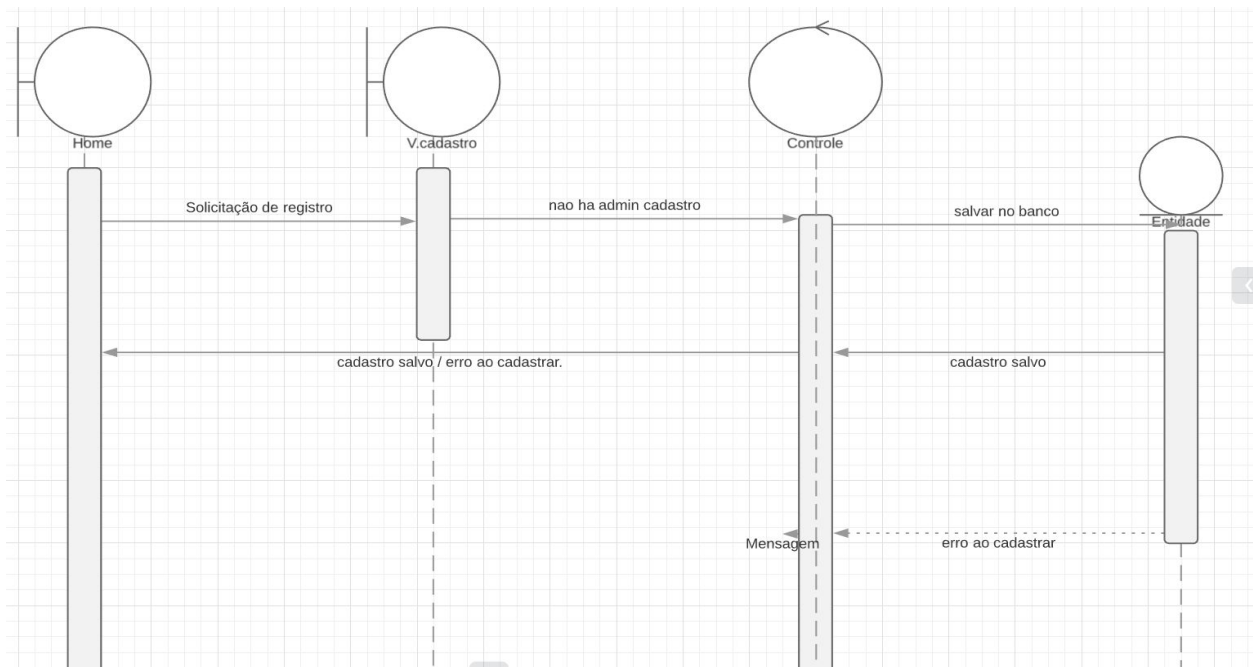
Fonte: próprio autor



**RF11 - Primeiro acesso**

<b>Função</b>	Realizar o cadastro do administrador ao primeiro acesso
<b>Descrição</b>	Acessa a tabela Usuario, onde possibilita o cadastro de administrador do sistema.
<b>Entradas</b>	As entradas correspondentes ao cadastro são o prontuário, nome, email, password e tipo, todas as entradas são String e não podem ser nulo, o password é limitado ao mínimo de 8 caracteres e possui sistema de criptografia em hashing, para manter a segurança e privacidade do usuário na comunicação com o banco de dados.
<b>Origem</b>	Administrador
<b>Saída</b>	Nenhuma.
<b>Destino</b>	-
<b>Ação</b>	O primeiro acesso será ativado somente quando não existir nenhum cadastro na tabela usuário.
<b>Pré-condição</b>	A tabela Usuario no Banco de dados não deve conter nenhum cadastro.
<b>Pós-condição</b>	O sistema deverá informar se obteve êxito no cadastro de Administrador ou não.
<b>Efeitos colaterais</b>	-

Figura 5– Diagrama de sequência do RF11



Fonte: próprio autor

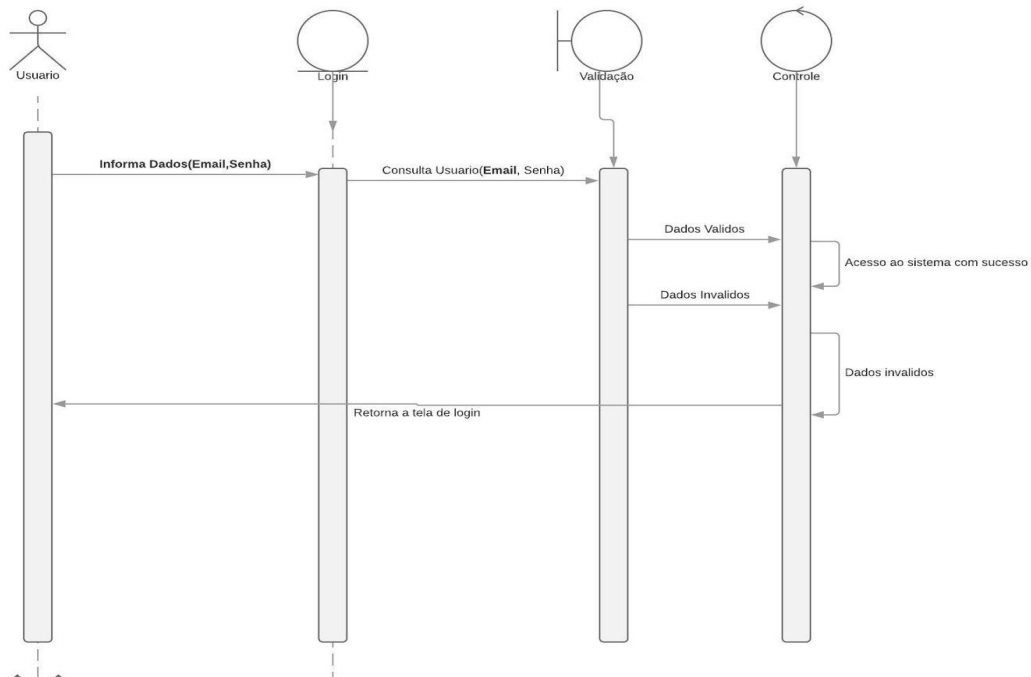
## RF12 - Login

<b>Função</b>	Realizar login no sistema
<b>Descrição</b>	Acessa a tabela Usuario onde possibilita realizar login no sistema como administrador ou coordenador do campus.
<b>Entradas</b>	Email e senha, ambos são entradas do tipo String e não podem ser nulas.
<b>Origem</b>	Administrador/Coordenador
<b>Saída</b>	Nenhuma.
<b>Destino</b>	-
<b>Ação</b>	Para acessar o sistema é necessário o usuário informar o email e senha, ao digitar o sistema acessa o banco de dados, onde é comparado a senha com a existente, caso as informações sejam validadas, consequentemente o usuário é logado no sistema e acessar a tela inicial.
<b>Pré-condição</b>	O usuário deve possuir cadastro no sistema.
<b>Pós-condição</b>	O sistema deverá informar se obteve êxito ou não no login.
<b>Efeitos colaterais</b>	-

Figura 6– Diagrama de sequência do RF12

RF12 - LOGIN

Vinicius 27/08/2019



Fonte: próprio autor

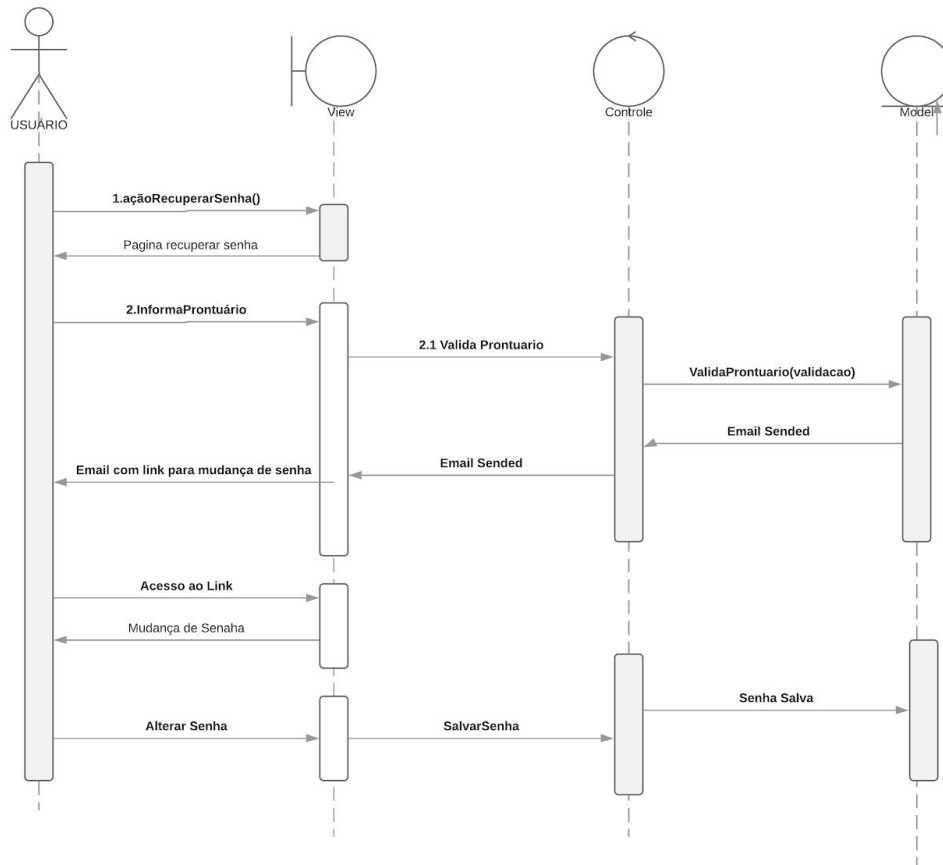
### RF13 - Recuperar senha

<b>Função</b>	Recuperar senha do usuário
<b>Descrição</b>	Envia um email para o usuário junto com um link para mudança de senha.
<b>Entradas</b>	Email
<b>Origem</b>	Administrador/coordenador
<b>Saída</b>	Envio de email
<b>Destino</b>	-
<b>Ação</b>	Para recuperar senha do usuário é necessário o usuário informar o email, o sistema irá verificar se esse prontuário existe no Banco de dados, caso seja confirmado será enviado um email, onde poderá acessar um link e por intermédio dele acessar uma tela de mudança de senha, posteriormente validada a senha fornecida por duas vezes, a mesma é alterada no banco de dados.
<b>Pré-condição</b>	O usuário deve possuir cadastro no sistema e um email válido
<b>Pós-condição</b>	O sistema deverá informar parte do email do qual foi enviado o e-mail para alteração da senha
<b>Efeitos colaterais</b>	-

Figura 7– Diagrama de sequência do RF13

RF-13 DIAGRAMA DE SEQUENCIA

Alan Landi | August 29, 2019



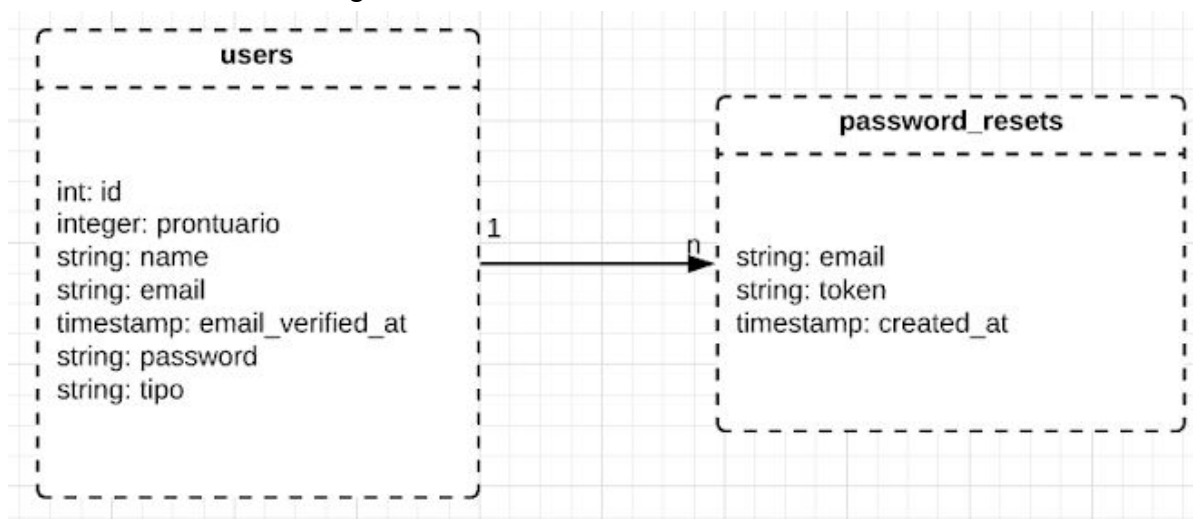
Fonte: próprio autor

## MODELO DO SISTEMA

Nessa seção descreve-se a forma de armazenamento dos dados pelo sistema.

Como ele utiliza o MySQL Server, na Figura 8 é exibido o modelo relacional implantado.

Figura 8 – Modelo relacional do sistema



Fonte: próprio autor

O dicionários de dados é uma forma de descrever as informações presentes no modelo relacional demonstrado na Figura 8. Abaixo iremos transcrever cada campo de cada tabela utilizadas no sistema.

### 1. Users

Armazena os usuários cadastrados no sistema.

<b>Campo</b>	ID
<b>Tipo de dado</b>	Inteiro (int)
<b>Pode ser nulo</b>	Não
<b>Chave primária</b>	Sim
<b>Chave estrangeira</b>	Não
<b>Outros</b>	Identificador (incrementando de forma crescente de 1 em 1)
<b>Descrição</b>	Armazena o identificador do usuário no sistema

<b>Campo</b>	Nome
<b>Tipo de dado</b>	String (50 caracteres)
<b>Pode ser nulo</b>	Não
<b>Chave primária</b>	Não
<b>Chave estrangeira</b>	Não
<b>Outros</b>	-
<b>Descrição</b>	Contém o nome do usuário do sistema.

<b>Campo</b>	Prontuário
<b>Tipo de dado</b>	String (9 caracteres)
<b>Pode ser nulo</b>	Não
<b>Chave primária</b>	Não
<b>Chave estrangeira</b>	Não
<b>Outros</b>	-
<b>Descrição</b>	Prontuário do usuário do sistema.

<b>Campo</b>	Email
<b>Tipo de dado</b>	String (50 caracteres)
<b>Pode ser nulo</b>	Não
<b>Chave primária</b>	Não
<b>Chave estrangeira</b>	Não
<b>Outros</b>	-
<b>Descrição</b>	Contêm o e-mail do usuário, também utilizado para recuperar senha.

<b>Campo</b>	password
<b>Tipo de dado</b>	String (Até 50 caracteres)
<b>Pode ser nulo</b>	Não
<b>Chave primária</b>	Não
<b>Chave estrangeira</b>	Não
<b>Outros</b>	Criptografia
<b>Descrição</b>	Armazena a senha criptografada do usuário do sistema.

<b>Campo</b>	Tipo
<b>Tipo de dado</b>	string(1)
<b>Pode ser nulo</b>	Não
<b>Chave primária</b>	Não
<b>Chave estrangeira</b>	Não
<b>Outros</b>	Primeiro acesso sempre será nível de acesso para administrador(A).
<b>Descrição</b>	Define o nível de acesso do usuário, podendo o mesmo ser administrador ou coordenador.

## 2. Password\_resets

Registra informações da solicitação de recuperação de senha.

<b>Campo</b>	Email
<b>Tipo de dado</b>	String (50 caracteres)
<b>Pode ser nulo</b>	Não
<b>Chave primária</b>	Não
<b>Chave estrangeira</b>	Não
<b>Outros</b>	-
<b>Descrição</b>	Contêm o e-mail do usuário, também utilizado para recuperar senha.

<b>Campo</b>	token
<b>Tipo de dado</b>	string(64)
<b>Pode ser nulo</b>	Não
<b>Chave primária</b>	Não
<b>Chave estrangeira</b>	Não
<b>Outros</b>	
<b>Descrição</b>	hash criada para validação da solicitação de troca de senha.

## CRONOGRAMA DE EXECUÇÃO

Incremento	Data de entrega	Requisito	Responsável
1	13/09/2019	RF11	Allan, Cleiton, Fabrício, Lucas, Vinícius.
1	13/09/2019	RF12	Allan, Cleiton, Fabrício, Lucas, Vinícius.
1	13/09/2019	RF13	Allan, Cleiton, Fabrício, Lucas, Vinícius.



## PLANEJAMENTO DE TESTES E MANUTENÇÃO

No decorrer do desenvolvimento da aplicação, é realizado testes de integração para encontrar erros de execução.

No final do desenvolvimento é realizado uma sequência de testes para validar cada requisito funcional do sistema tanto de unidade quanto para verificar o correto funcionamento do sistema.

Após os testes validando cada requisitos, serão feito os testes de releases para a fim de encontrar possíveis brechas no desenvolvimento.

### Usuário

- Tentativa de inserção do nome do usuário com mais caracteres do que o especificado no banco de dados;
- Tentativa de inserção do prontuário do usuário com mais caracteres do que o especificado no banco de dados;
- Tentativa de inserção do email do usuário com mais caracteres do que o especificado no banco de dados;
- Tentativa de inserção do password do usuário com mais caracteres do que o especificado no banco de dados;
- Tentativa de cadastro de usuário com campos não nulos sem preenchimento, dentre eles estão nome, prontuário, email e senha;
- Tentativa de cadastro de usuário com o banco de dados sem conexão.

### Login

- Tentativa de login com banco de dados sem conexão;
- Tentativa de login com usuário não cadastrado no sistema;
- Tentativa de acesso via url, a um endereço sem permissão concedida;

### Recuperar senha

- Tentativa de recuperar senha através de um email de usuário não cadastrado.
- Tentativa de modificar senha após o tempo limite para mudança especificado no sistema se esgotar;