

# Especificação de Requisitos não-funcionais do Sistema

## Introdução

- Nesse documento, serão apresentados requisitos não-funcionais para assegurar a integridade e qualidade (1) de funcionamento do produto de software a ser desenvolvido, assim como requisitos de interface (2), regras de negócio (3), restrições (4), conformidades (5) e documentação a ser produzida (6) durante a implementação do mesmo.

## 1. Requisitos de Qualidade do Sistema

### 1.1. Usabilidade

- O usuário deve ser capaz de reconhecer o propósito do software e a razão para utilizá-lo.
- O software deve ser facilmente utilizado por usuários que possuam familiaridade básica com interfaces web.
- O software deve apresentar uma interface gráfica de fácil entendimento, contando com escritas e funcionalidades explícitas.
- O software deve apresentar telas simples e objetivas, tendo foco nas informações necessárias para realização das tarefas.
- O usuário deve conseguir realizar a tarefa de criar conta em até 5 minutos.
- O usuário deve conseguir efetuar login em um prazo de até 30 segundos.
- O usuário deve conseguir criar e especificar quadros de maneira ágil e intuitiva.
- O usuário deve conseguir adicionar e realocar cartões de forma rápida, simples e intuitiva.
- O software deve possuir títulos que identifiquem as telas, permitindo a orientação do usuário durante o uso.
- O software deve conter alertas de confirmação das operações, facilitando a compreensão do usuário acerca da operação realizada.
- O usuário deve ser capaz, mesmo após determinado tempo ser uso do sistema, conseguir utilizar as funcionalidades básicas oferecidas.
- O software deve oferecer variação visual com o propósito de facilitar a usabilidade de usuários com alguma limitação visual.
- O software deve oferecer a possibilidade de assistência sonora para usuários com limitação visual e motora.

### 1.2. Confiabilidade

- O software deve, em condições normais, oferecer plena execução das funcionalidades descritas anteriormente.
- O software deve oferecer funcionalidade a momento e em toda localidade.
- O software deve poder ser acessado de todo navegador web que ofereça suporte à navegação.
- O software deve armazenar temporariamente os dados em caso de falha da conexão de rede, para posteriormente efetuar atualização do banco de dados após reestabelecimento da conexão.
- O software deve ser capaz de armazenar e recuperar dados perdidos decorrente de falhas do sistema ou da rede.

- O software deve ser capaz de recuperar os dados perdidos por falhas em um prazo de até 30 segundos, não prejudicando assim a experiência do usuário.

### **1.3. Eficiência de Desempenho**

- O software deve apresentar tela de login em até 10 segundos.
- O software deve efetuar o registro de um quadro no prazo de até 5 segundos.
- O usuário deve conseguir realizar consulta de um quadro em um prazo de até 5 segundos.
- O software deve efetuar a criação e registro de um cartão em um prazo de até 5 segundos.
- O usuário deve conseguir realizar consulta e movimentação de cartões em um prazo de até 10 segundos.
- O software deve efetuar a exclusão de um quadro em um prazo de até 10 segundos.
- O software deve promover troca de telas em até 5 segundos, com a finalidade de otimizar a experiência do usuário.
- O software deve ser capaz de suportar a criação e alocação de (definir) quadros e cartões por usuário.

### **1.4. Capacidade de Manutenção**

- Seguir as boas práticas definidas na documentação da arquitetura, como: Modularização, testes unitários, etc.

### **1.5. Portabilidade**

- O software deve apresentar funcionamento pleno em qualquer navegador web.
- Deve ser possível a utilização do software em desktops, mobiles e todos os outros modelos de hardware que tenham suporte à navegação web.
- É necessário para a utilização do software a instalação de um navegador web da preferência do usuário.
- O software não necessita de instalação prévia, podendo ser utilizado totalmente de forma "on-line".
- O software deve apresentar as mesmas funcionalidades independente do sistema de hardware utilizado.

### **1.6. Segurança**

- O software deve possuir sistema de autenticação de usuários.
- O software deve armazenar os dados de forma criptografada.
- O software deve assegurar restrição de acesso aos usuários.
- O software deve promover diferenciação de entidades, permitindo a atribuição de funcionalidades somente à entidades determinadas.
- O software deve conter alguma forma de registro de atividades, garantindo assim a integridade dos dados.
- O software deve contar com restrições de modificação.
- O software deve seguir as normas definidas pela Lei 13.709/18 - Lei de Proteção de Dados.

### **1.7. Compatibilidade**

- O software deve apresentar pleno funcionamento quando integrado a um ambiente de desenvolvimento já estabelecido.

- O software não deve interferir no funcionamento de outros produtos de software usados concomitantemente.
- Ao ser integrado com outros produtos de software, deve ser possível ao sistema de software aqui desenvolvido estabelecer conexão e troca de informações de maneira controlada.

## **2. Requisitos de Interface**

### **2.1. Interface de Usuário**

#### **Aspectos Visuais**

- O sistema deve ter interface limpa e amigável ao usuário. São esperadas letras grandes com títulos de identificação das telas, apresentando foco na funcionalidade oferecida em determinada tela. São esperados botões com títulos objetivos para que o usuário se localize e tenha certeza da funcionalidade que está sendo usada.

#### **Requisitos de Layout e Navegação**

- O sistema deve contar com informações apresentadas no centro da tela acerca da funcionalidade que está sendo utilizada.
  1. Na tela de login, são apresentados os campos para efetuar a autenticação no centro da tela.
  2. Na tela de cadastro, são apresentados os campos para a realização do cadastro no centro da tela e na parte direita superior está presente a opção de efetuar login para usuário já cadastrado. Após realizado cadastro, é exibido um alerta de confirmação com a finalidade de orientar o usuário.
  3. Ao efetuar o login, o usuário é redirecionado na tela de “Área de trabalho”. Nessa tela, estão presentes todos os quadros em que o usuário está participando. Na parte superior esquerda, está presente um menu que apresenta algumas opções de navegação pelo sistema.
  4. Ao clicar em algum quadro, é apresentado ao usuário uma tela com todos os cartões presentes naquele quadro. Nessa tela, é possível utilizar de funcionalidades como a adição de cartões, a troca de colunas e a alteração de informações.
  5. Deve haver no sistema uma tela de alteração de informações para o usuário, contando com campos para cadastro de novas informações.

#### **Consistência**

- Com o intuito de facilitar a navegação e orientação do usuário, o sistema deve conter títulos bem destacados em cada tela, trazendo foco à funcionalidade oferecida em determinada tela. Junto a isso, o sistema deve apresentar alertas de confirmação após realizada alguma operação, garantindo assim ao usuário a integridade da operação realizada. As informações principais devem sempre aparecer ao centro da tela e as funcionalidades devem seguir um fluxo lógico de operação, fazendo com que o usuário entenda seu funcionamento após pouco tempo de uso.

#### **Requisitos de Personalização do Usuário**

- O sistema deve conter opções personalizadas ao usuário. Deve ser possível ao usuário somente visualizar os quadros em que ele está cadastrado, além de só ser possível a exclusão de um quadro criado por ele mesmo. Além disso, por se tratar de um sistema web, devem ser guardadas informações de preferência do usuário, como o padrão de cor adotado.

## 2.2. Interfaces para sistemas externos ou dispositivos

### Interfaces de Software

- Por se tratar de um SaaS (Software as a Service – Software oferecido como um Serviço), o sistema necessita de um navegador web oferecido por terceiros para a utilização de suas funcionalidades.

### Interfaces de Hardware

- Por se tratar de um Software que executa em plataforma web, não há restrições de Hardware para a utilização do sistema. Os requisitos de Hardware são somente os necessários para a utilização do navegador web de preferência do usuário, sendo possível o pleno funcionamento de todas as funcionalidades do produto de software.

### Interfaces de comunicação

- É necessário para a utilização do sistema uma conexão de rede, sendo possível o acesso à Internet.

## 3. Regras de negócios

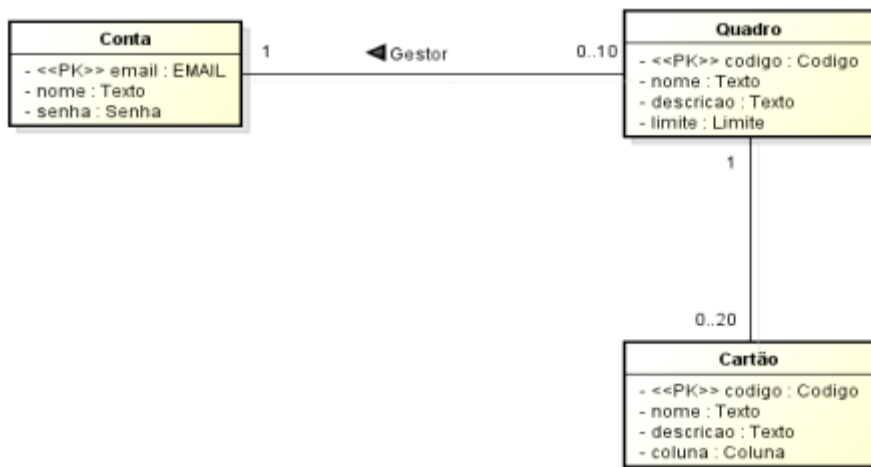
### 3.1. Regras para cadastramento de dados:

1. O domínio "CODIGO" deve possuir formato LLDD, em que L é uma letra maiúscula (A - Z) e D é um dígito (0 - 9).
2. O domínio "COLUNA" só assume os valores "SOLICITADO", "EM EXECUCAO", "CONCLUIDO".
3. O domínio "EMAIL" deve possuir formato *nome@domínio*, em que *nome* é composto por 2 a 10 caracteres, *domínio* é composto por 2 a 10 caracteres e cada caractere é uma letra (A - Z, a - z), dígito (0 - 9) ou ponto (.). O caractere "@" não pode ser imediatamente precedido ou sucedido por ponto (.) e não há pontos (.) em sequência.
4. O domínio "LIMITE" só assume os valores "5", "10", "15", "20".
5. O domínio "SENHA" deve possuir formato XXXXX, em que X é caractere que consiste em uma letra (A - Z, a - z), dígito (0 - 9) ou sinal de pontuação (., ; ? !). Pelo menos um caractere é letra maiúscula, pelo menos um caractere é letra minúscula, pelo menos um caractere é dígito, pelo menos um caractere é um sinal de pontuação e não há caractere duplicado.
6. O domínio "TEXTO" possui de 5 a 30 caracteres, em que cada caractere consiste de uma letra (A - Z, a - z), dígito (0 - 9), sinal de pontuação (., ; ? !) ou espaço em branco. Não há espaços em branco em sequência, não há sinais de pontuação em sequência, não há acentuação, o primeiro caractere é uma letra maiúscula e o primeiro caractere após sinal de pontuação (exceto vírgula e ponto-e-vírgula) é letra maiúscula.

### 3.2. Regras de integridade:

1. Usuário só pode eliminar quadro por ele criado.
2. Eliminar quadro resulta na eliminação dos cartões associados.

3. Número máximo de cartões na coluna trabalho em progresso é igual ao limite do trabalho em progresso (Work in progress limit) informado no quadro.
4. O sistema deve também assegurar as regras representadas no seguinte diagrama:



## 4. Restrições do Sistema

- O software necessita de um navegador web oferecido por terceiros para a utilização.
- O software necessita de uma conexão de rede para a utilização.
- O software deve ser desenvolvido utilizando a linguagem Java 11 ou superior para a implementação do Backend.
- O software deve utilizar Spring Framework 5 ou superior no desenvolvimento do Backend.
- O software deve utilizar o SGBD MySQL para a implementação do Banco de Dados.
- O software deve utilizar HTML 5 para a implementação do Frontend.
- O software deve utilizar CSS 3 para a implementação do Frontend.
- O software deve utilizar Typescript (Superbiblioteca de Javascript) para a implementação do Frontend.
- O software deve utilizar o Framework Angular 2 na versão 11 ou superior para a implementação do Frontend.
- O software deve ser implementado de forma que possa oferecer suas funcionalidades em todo navegador web, independente do Hardware que o navegador está sendo utilizado.

## 5. Conformidades do Sistema

### 5.1. Requisitos de Licenciamento

- O software aqui desenvolvido visa a utilização corporativa e otimização do processo de desenvolvimento na empresa, sendo fundamental seu licenciamento no processo de implementação para controle de desenvolvimento em larga escala com fins lucrativos.

### 5.2. Avisos Legais

- O software aqui desenvolvido não se responsabiliza por falhas decorrentes de implementação de terceiros no processo de desenvolvimento ao utilizar a ferramenta. Toda e qualquer falha ocasionada por terceiros, que não possuam relação direta à utilização do software não são caracterizadas como responsabilidade do produto de software kanban.

- A utilização correta do software é tarefa do usuário, qualquer erro ocasionado por mal uso do sistema será contabilizado como responsabilidade do próprio usuário.
- A utilização do software em ambientes que ele não foi desenvolvido para executar poderá acarretar em falhas, não sendo responsabilidade do produto de software.

### **5.3. Padrões Aplicáveis**

- O software acima descrito deve ser desenvolvido utilizando requisitos de qualidade descritos na norma "ISO/IEC 25010" (<https://iso25000.com/index.php/en/iso-25000-standards/iso-25010>), garantindo assim nível de qualidade aceitável e pleno funcionamento das funcionalidades propostas.
- Deve ser utilizada arquitetura "MVC" para a implementação do sistema. Para mais informações, segue link informativo e introdutório (<https://www.geeksforgeeks.org/mvc-framework-introduction/>).

## **6. Documentação do Sistema**

- O software deve contar com documentação on-line disponível para o usuário a qualquer momento. Na documentação (feita preferencialmente pela parte da equipe responsável pelos testes do sistema), devem ser explicitadas as funcionalidades do sistema e demonstrada de forma clara e objetiva como e porque usá-las. Deve haver também na documentação a demonstração de cenários que possam vir a acontecer e suas possíveis resoluções.
- O software também deverá contar com documentação para desenvolvedores (gerada pelos próprios desenvolvedores do sistema), possibilitando assim manutenção e atualização posterior.