



**UNIVERSIDADE FEDERAL DO ACRE**  
**CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS**  
**CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**Documento de Requisitos de Software**

**Aplicação Web de Gerenciamento de Atividades**  
**Versão 0.3**

**Desenvolvedores:**

**Gabriel Martins**  
**Clécio Elias**  
**Matheus Diógenes**

**Rio Branco – AC**  
**2022**

### Histórico de Alterações

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>
25/01/2023	0.0	Elaboração do documento de requisitos	Gabriel Martins
25/01/2023	0.1	Revisão e adições no documento de requisitos	Clécio Elias
25/01/2023	0.2	Revisão e finalização da versão	Matheus Diógenes
30/01/2023	0.3	Revisão e adição de novas diretrizes	Matheus Diógenes
22/03/2023	0.4	Revisão e modificação de requisitos	Clécio Elias

## **1. Análise do Problema**

O professor Doutor Daricélio Soares, observou que uma entidade tinha a necessidade de controlar as atividades exercidas e para criar registros para o futuro. Diante desse cenário, ele apresentou essa demanda aos seus estudantes para que os mesmos tivessem condições de implementar um modelo de sistema CRUD(Create, Read, Update, Delete é um acrônimo para as maneiras de se operar em informação armazenada) utilizando desenvolvido orientado a reuso. Os estudantes fizeram uma análise prévia dos requisitos e dentro da realidade se observou ser possível a implementação.

Este documento especifica as funcionalidades e restrições de uma Aplicação Web Para Gerenciamento de Atividades, dando uma visão das características gerais do sistema e seus requisitos funcionais e não funcionais.

### **1.1 Identificação dos Requisitos**

Por convenção, a referência a requisitos é feita através do nome da subseção onde eles estão descritos, seguidos do identificador do requisito, de acordo com a especificação a seguir: [nome da subseção, identificador do requisito]. Por exemplo [Adicionar atividades.RF001] se refere ao requisito funcional da subseção Adicionar Atividades com identificador de requisito [RF001].

### **1.2 Definição de Prioridades**

#### **1.2.1 Essencial**

Requisitos indispensáveis e que afetam diretamente no bom funcionamento do sistema, tornando-os obrigatórios e máxima prioridade.

#### **1.2.2 Importante**

Requisitos sem os quais o sistema pode funcionar corretamente, porém não de forma totalmente satisfatória, portanto, sendo de prioridade moderada.

#### **1.1.3 Desejável**

Funcionalidades extras que agregam ao sistema, mas não possuem impacto no seu funcionamento básico, resultando em uma menor prioridade.

## **2. Necessidades Básicas do Cliente**

As necessidades de um grupo específico de partes interessadas em relação à solução proposta. Esse tópico é conhecido como requisitos funcionais. Eles podem ser usados para descrever como um conjunto particular de usuários de uma solução irão interagir com ela e como um produto irá atender as necessidades de diferentes grupos de clientes.

## **3. Estudo de Viabilidade**

A análise de viabilidade de projetos é um tipo de estudo que antecede as etapas de planejamento de um software. Neste momento, o gestor avaliará, a partir de uma longa análise de dados, se é realmente viável o investimento necessário para criar e distribuir o sistema.

### **3.1. Viabilidade Técnica**

Viabilidade técnica é uma avaliação da praticidade de uma solução técnica específica e a disponibilidade dos recursos técnicos e dos especialistas. O sistema é tecnicamente viável para o desenvolvimento, pois está dentro da capacidade dos desenvolvedores.

### **3.2. Viabilidade Econômica**

É uma avaliação de custo-eficiência de um projeto ou solução. Conhecida como análise de custo-benefício. Tão logo os requisitos específicos e soluções sejam identificados, o analista pode levar em consideração os custos e benefícios de cada alternativa. O projeto é economicamente viável diante da sua problemática.

### 3.3. Viabilidade Legal

O projeto é analisado do ponto de vista da legalidade. Isso inclui a análise de barreiras de implementação legal do projeto, atos de proteção de dados ou leis de mídia social, certificado de projeto, licença, direitos autorais. O projeto dentro da legalidade é viável, tendo em vista as partes envolvidas.

### 4. Missão do Software

As operações modelo de sistema proposto(CRUD) são amplamente utilizadas em muitos aplicativos suportados pelo banco de dados relacional subjacente. Esses quatro recursos principais de CRUD(CREATE, READ, UPDATE, DELETE) são muito comuns no suporte a uma variedade de funções-chave em diferentes modelos de negócios e verticais do setor. O software vai promover maior segurança no armazenamento de informações da entidade.

### 5. Limites do Sistema

ID	Funcionalidade	Justificativa
L1	Gerenciamento de atividades	O software tem um objetivo de controlar as atividades da entidade e não pode ter sua função principal modificada

### 6. Benefícios Gerais

ID	Benefício
B1	Uso do modelo relacional e não relacional de banco de dados
B2	Maior segurança para armazenar dados
B3	Facilidade para manipular os dados
B4	Baixa demanda computacional
B5	Praticidade no desenvolvimento do código e das regras de negócio

## 7. Restrições

ID	Restrição	Descrição
R1	Sistema legado	requer maior esforço na adequação das regras de negócio
R2	Requer atenção na execução dos comandos	<i>UPDATE</i> e <i>DELETE</i> , quando executados diretamente no banco de dados tem de ser de forma segura

## 8. Atores

ID	Atores	Descrição
A1	Usuários da Entidade	Estão diretamente envolvidos, já que farão uso do sistema
A2	Entidade que encomendou o sistema	Os proprietários do sistema que financiaram seu desenvolvimento
A3	Equipe de Desenvolvimento	Responsável direta pelo desenvolvimento e manutenção do sistema

## 9. Requisitos Funcionais

ID	Funcionalidade	Necessidades	Classificação
RF1	Adicionar Atividades	Possibilidade de adicionar título, descrição, data de início, fim e prioridade.	Essencial
RF2	Adicionar imagens, textos e editá-los nas atividades	Possibilidade de editar os detalhes de uma atividade já adicionada, incluindo título, descrição, data de início e fim, e prioridade.	Essencial
RF3	Adicionar subatividades dentro de uma atividade	Possibilidade de adicionar subatividades dentro de uma atividade principal, formando uma hierarquia de atividades.	Essencial

<b>RF4</b>	Possuir uma agenda com calendário	Possibilidade de visualizar as atividades em um calendário, com possibilidade de filtrar por data e prioridade, além de agendar atividades para o futuro.	Importante
<b>RF5</b>	Opção para adicionar notas	Possibilidade de adicionar "sticky notes" ou "notas de rodapé" tanto nas atividades/subatividades quanto nos dias do calendário	Importante
<b>RF6</b>	Possuir uma hierarquia de atividades	Possui uma hierarquia visual de atividades e subatividades, podendo abri-las para visualização e/ou edição das mesmas.	Importante
<b>RF7</b>	Opção para adicionar arquivos prontos	Possibilidade de adicionar relatórios, atividades, evidências e ou arquivos já prontos na plataforma.	Importante
<b>RF8</b>	Gerar relatórios sobre as atividades	Possibilidade de gerar relatórios sobre as atividades, incluindo informações como título, descrição, data de início e fim, prioridade, além da opção de editar o relatório gerado já que o template	Importante

		pode ser relativo.	
--	--	--------------------	--

## 10. Requisitos Não-Funcionais

ID	Requisitos	Categoria	Classificação
<b>NRF1</b>	O sistema deve ter uma boa performance de velocidade.		Essencial
<b>NRF2</b>	O sistema deve ser desenvolvido de forma a ser fácil de manter e atualizar.		Essencial
<b>NRF3</b>	O sistema deve ser desenvolvido com usabilidade e acessibilidade em mente.		Essencial
<b>NRF4</b>	O sistema deve ser seguro e prover tratativa de erros.		Essencial
<b>NRF5</b>	O sistema deve ser desenvolvido como uma plataforma web.		Essencial



## **11. Requisitos de Hardware**

### **11.1. Configuração Mínima**

Navegador com acesso à rede

## **12. Ferramentas de Desenvolvimento e Licença de Uso**

- a. Utilizar um padrão de arquitetura de software consistente e bem estabelecido, como o MVC (Model-View-Controller).
- b. Utilizar uma metodologia de desenvolvimento de software, como o Scrum ou o Kanban.
- c. Utilizar um controle de versão, como o Git, para garantir que as alterações no código sejam rastreadas e revertidas caso necessário, além de ter em registro todo o desenvolvimento do software (github do projeto: [https://github.com/Cleps/plataforma\\_web-EngSoftw](https://github.com/Cleps/plataforma_web-EngSoftw) ).
- d. Utilizar uma estratégia de gerenciamento de dependências para garantir que as bibliotecas e frameworks utilizados no projeto sejam atualizados de forma controlada e segura.