

**ENGENHARIA DE REQUISITOS E ANÁLISE DE SISTEMAS**

**DOCUMENTAÇÃO DO SISTEMA DE CADASTRO WEB - FASE 1**

**ALUNOS:**

**CINTIA MENDES BERNARDO**

**JHEAN MONTEIRO DA SILVA**

**LORENZO SAMPAIO DE GUADELUPE**

**TURMA A**

**PROF. GIOLIANO BERTONI**

**SAQUAREMA**

**2025**

## Trabalho pedido pelo professor

O sistema deve utilizar obrigatoriamente as seguintes tecnologias

- **Frontend:** HTML5, CSS3, JavaScript
- **Backend:** Python 3 + Flask
- **Banco de Dados:** SQLite3

Inicialmente estão disponíveis os arquivos encontrados na pasta CADASTRO WEB

Atividade

### 1. FASE 1 – DOCUMENTAÇÃO + FRONTEND

- Fazer o protótipo funcionar ✓
  - Corrigindo os prováveis erros ✓
- Propor melhorias ao projeto de Cadastro de Usuários Incluindo mais campos necessários ✓
- Realizar a documentação do sistema ✓
  - Requisitos Funcionais e Não Funcionais ✓
  - Diagrama Entidade Relacionamento ✓
  - Diagrama de Caso de Uso ✓
- Apresentação (06/11) + relatório ✓

### 2. FASE 2 – BACKEND + BD

- Fazer uma pesquisa sobre o FLASK e o SQLite
- Fazer funcionar o sistema integrando o Backend ao BD
- Apresentação (13/11) + relatório

## OBSERVAÇÃO

O objetivo é que tente inicialmente rodar a aplicação localmente. Mas caso tenha problemas relacionados a instalação no laboratório, como alternativa:

- <https://render.com/>
- <https://www.pythonanywhere.com/>

ENGENHARIA DE REQUISITOS E ANÁLISE DE SISTEMAS	1
DOCUMENTAÇÃO DO SISTEMA DE CADASTRO WEB - FASE 1	1
ALUNOS:	1
CINTIA MENDES BERNARDO	1
JHEAN MONTEIRO DA SILVA	1
LORENZO SAMPAIO DE GUADELUPE	1
TURMA A	1
PROF. GIOLIANO BERTONI	1
SAQUAREMA	1
2025	1
1. Introdução	4
1.1. Correção do Protótipo	4
1.2. Melhorias Propostas	4
1.3. Documentação	4
2. Correção do Protótipo	4
2.1. Erros encontrados	4
3. Melhorias (Campos e Funcionalidades)	5
3.1. Campos Atuais:	5
3.2. Melhorias Sugeridas:	5
4.Documentação do Sistema	5
4.1. Requisitos Funcionais (RF)	5
4.2. Requisitos Não Funcionais (RNF)	6
3. Diagrama Entidade Relacionamento	7
4. Diagrama de Caso de Uso	8
5. Conclusão	8

## 1. Introdução

Este relatório apresenta os resultados da Fase 1 do projeto de desenvolvimento de um sistema de cadastro de usuários, utilizando **HTML5**, **CSS3** e **JavaScript** no frontend, conforme exigido. A Fase 1 teve como objetivos:

### 1.1. Correção do Protótipo

Encontrar erros.  
Corrigir códigos.  
Implementar funcionalidades.

### 1.2. Melhorias Propostas

Novos campos.  
Validações adicionais.  
UX/UI.

### 1.3. Documentação

Requisitos Funcionais.  
Requisitos Não Funcionais.  
DER (Imagem).  
Diagrama de Caso de Uso (Imagem).

## 2. Correção do Protótipo

### 2.1. Erros encontrados

Arquivo	Erro	Correção Aplicada
cadastro.html	Tag <input> escrita errada.	Corrigido para <input>.
cadastro.html	href="stile.css" escrito errado.	Corrigido para Href="style.css".
cadastro.html	Botão de submit ausente.	Adicionado <button type="submit">.

cadastro.html	Sem validação de CPF.	Integrada função ValidarCPF().
cadastro.html	Apenas 2 campos Nome e E-mail.	Adicionados RG, CPF, Telefone, Senha.

**Resultado:** Protótipo 100% funcional.

### 3. Melhorias (Campos e Funcionalidades)

#### 3.1. Campos Atuais:

- Nome e Email.

#### 3.2. Melhorias implementadas:

CAMPO	JUSTIFICATIVA
Nome	(Max. 50 caracteres).
RG	Documento oficial (00.000.000-0).
CPF	Melhora a usabilidade (000.000.000-00) com validação real.
Telefone	Número para contato ((00) 00000-0000).
Email	Email para contato (cadastro@gmail.com)
Confirmação de Senha	(Min. 6 caracteres, oculta).

Funcionalidades	Descrição
Enter passa de campo	Aperte enter e pula pro próximo campo.
Máscaras em tempo real	RG, CPF e telefone formatam enquanto digita.
Toast flutuante	Avisa quando atingir 50 caracteres no nome.
Feedback visual	Mensagem verde (sucesso) ou vermelha (erro).
Limpeza automática	Após sucesso, limpa todos os campos.
UX profissional	Cores contrastantes, sombras, hover, foco brilhante.

## 4.Documentação do Sistema

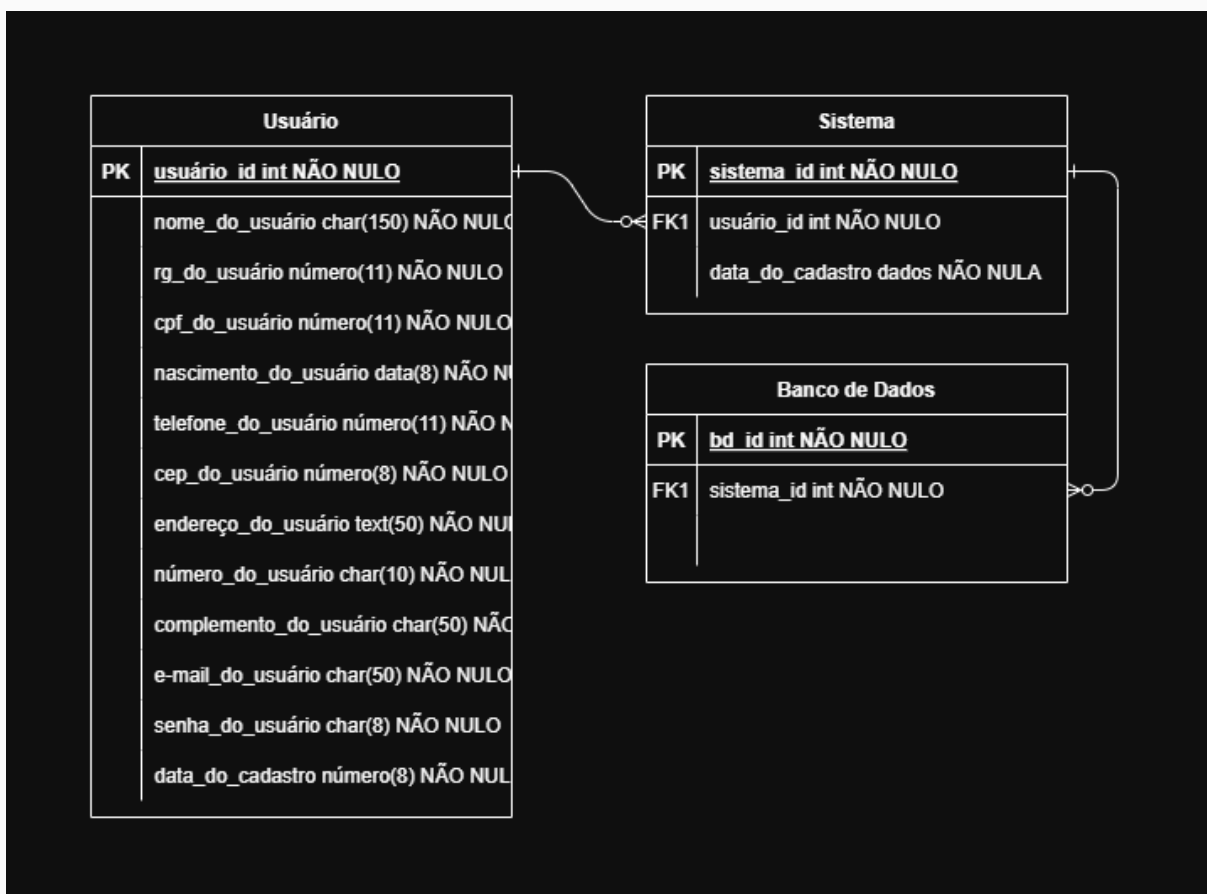
### 4.1. Requisitos Funcionais (RF)

ID	Descrição
RF001	O usuário deve preencher um formulário de cadastro.
RF002	O sistema deve validar todos os campos obrigatórios.
RF003	O sistema deve validar o CPF com o algoritmo oficial.
RF004	O sistema deve aplicar máscaras automáticas em RG, CPF e Telefone.
RF005	O sistema deve permitir navegação por Enter.
RF006	O sistema deve exibir feedback visual de sucesso ou erro.
RF007	O sistema deve limpar o formulário após cadastro bem-sucedido.
RF008	O sistema deve impedir cadastro com dados inválidos.

### 4.2. Requisitos Não Funcionais (RNF)

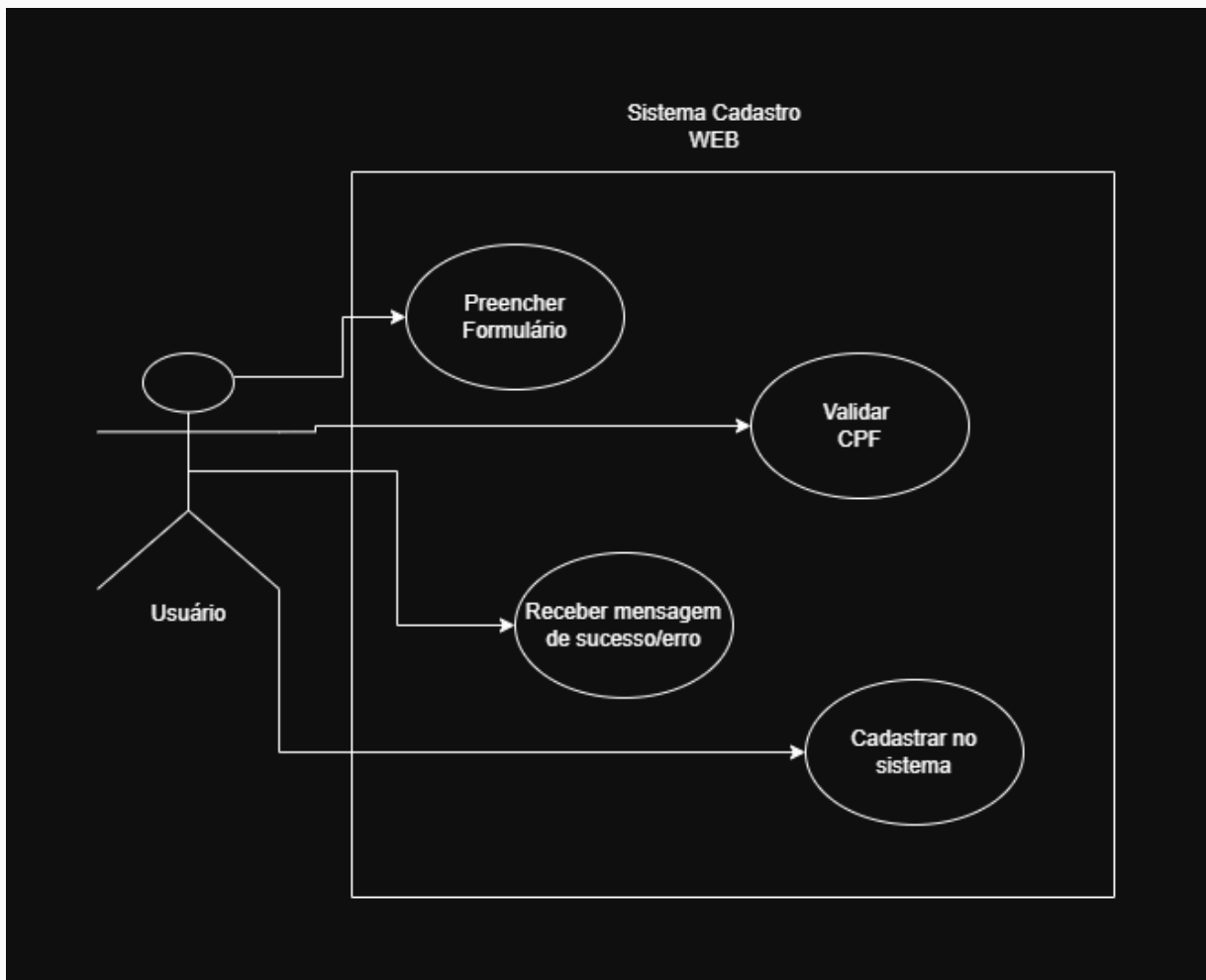
ID	Descrição
RNF001	Interface em Português BR.
RNF002	Responsivo (mobile-friendly).
RNF003	Validação no lado do cliente (JS).
RNF004	Segurança (senhas ocultas).
RNF005	Código limpo, comentado e organizado.
RNF006	Sem dependências externas.
RNF007	Cores de alto contraste.

### 4.3. Diagrama Entidade Relacionamento





#### 4.4. Diagrama de Caso de Uso



## 5. Conclusão

### 5.1. Fase 1 foi concluída com 100% de sucesso

Protótipo 100% Funcional.  
Validação de CPF integrada.  
Layout moderno e responsivo.  
Documentação completa.  
Pronto para integração com **Flask** + **SQLite3** na fase 2.

### 5.2. Próximos passos (13/11):

Criar backend com Flask  
Integrar com banco SQLite3  
Salvar dados permanentemente



## 6. ANEXOS

1. `index.html` (código completo)
2. `style.css` (estilos)
3. `script.js` (validações e máscaras)
4. Prints do sistema funcionando
5. Arquivos .drawio do DER e Caso de Uso

**SAQUAREMA, 05 de novembro de 2025**

---

**Cintia Mendes Bernardo**

---

**Jhean Monteiro da Silva**

---

**Lorenzo Sampaio de Guadalupe**

