

## LEVANTAMENTO DE REQUISITOS

### REPOSITÓRIO INSTITUCIONAL DE DOCUMENTOS ESCOLARES - RIDE

#### 1. INTRODUÇÃO

O presente documento descreve os requisitos funcionais e não funcionais, assim como o fluxo de operação de um software para a web, que funcionará como um Repositório Institucional de Documentos Escolares - RIDE, na web.

O objetivo do software é permitir que a equipe de gestão, diretores, coordenadores e secretaria escolar, disponibilize online modelos de documentos que possam ser baixados pelos professores, preenchidos e reenviados.

O software também deverá manter um controle dos documentos recebidos, com permissões distintas para cada perfil de usuário (diretores, coordenadores, secretaria escolar e professores).

O software será desenvolvido para a web usando:

- Python com Flask (Framework) → **Back-end**.
- HTML, CSS e JavaScript (Bootstrap) → **Front-end**.
- MySQL → **Banco de dados**.

#### 2. REQUISITOS FUNCIONAIS

Os requisitos funcionais descrevem o que o software deve fazer.

##### 2.1. CADASTRO DE USUÁRIOS

- O software deve permitir que o Diretor cadastre novos usuários com os perfis de Gestor e Professor.
- O software deve permitir que o Gestor cadastre novos usuários com o perfil de Professor”.
- O software não permitirá que os usuários alterem sua própria senha.
- O software deve permitir que:
  - O Diretor redefina as senhas dos Gestores.
  - O Gestor redefina as senhas dos Professores.

## **2.2. AUTENTICAÇÃO DE USUÁRIOS (LOGIN)**

- O software deve exigir autenticação de usuário para acesso.
- O software deve manter um controle de sessão de usuários.

## **2.3. CATEGORIZAÇÃO DE DOCUMENTOS**

- O software deve permitir que os perfis de Diretor e Gestor cadastrem categorias de documentos, como: Plano de Ensino, Formulários e Requerimentos, em geral.

## **2.4. ENVIO DE MODELOS DE DOCUMENTOS**

- O software deve permitir que o Diretor e os Gestores enviem modelos documentos para o repositório, de modo que os professores tenham acesso para baixar e preencher, de acordo com as necessidades.
- O software deve permitir o envio de arquivos em formato editável (DOCX).
- O software deve armazenar as seguintes informações:
  - ID do Documento (Gerado Automaticamente).
  - Título.
  - Descrição.
  - Categoria.
  - Nome do Usuário que Enviou.
  - Data de Envio.

## **2.5. VISUALIZAÇÃO E DOWNLOAD DE DOCUMENTOS**

- O software deve permitir que os Professores visualizem a lista de modelos de documentos disponíveis e separados por categoria.
- O software deve permitir que os Professores façam download dos modelos de documentos para preenchimento.

## **2.6. ENVIO DE DOCUMENTOS PREENCHIDOS**

- O software deve permitir que o Professor envie um documento preenchido relacionado a um modelo de documento.
- Caso o Professor envie novamente o mesmo documento preenchido, o software deve sobrescrever o envio anterior.
- O software deve registrar:

- ID do Envio de Documento.
- Modelo de Documento Relacionado.
- Usuário (Professor).
- Arquivo Preenchido.
- Data do Envio.

## 2.7. VISUALIZAÇÃO DE DOCUMENTOS PREENCHIDOS

- O software deve permitir que os Gestores visualizem os documentos preenchidos enviados pelos Professores.
- O software deve permitir que o Diretor visualize todos os documentos preenchidos, incluindo os enviados pelos Professores e publicados pelos Gestores.

## 3. REQUISITOS NÃO FUNCIONAIS

Os requisitos não funcionais especificam características e restrições do software.

- O software será construído em **Python 3.x** com o framework **Flask**.
- A interface gráfica de usuário será construída com **Bootstrap**, uma biblioteca de HTML, CSS e JavaScript com foco em usabilidade e responsividade.
- Os dados serão armazenados em um banco de dados **MySQL**.
- O software deverá permitir upload e download de arquivos.
- O código deve seguir boas práticas de modularização e organização.
- O software deverá ter controle de sessões, acesso e autenticação de usuário com senha.

## 4. FLUXO DE USO

1. O usuário acessa o software e, logo na página inicial, realiza o login com seu CPF e senha.
2. O software identifica o perfil do usuário e exibe as opções disponíveis de acordo com as permissões em uma página gerencial.
3. As principais funcionalidades gerenciais são:
  - Para Diretores:
    - Cadastrar gestores e professores.
    - Redefinir senhas de gestores e professores.

- Cadastrar categorias de documentos.
  - Enviar modelos de documentos.
  - Visualizar modelos de documentos.
  - Visualizar documentos preenchidos.
- Para Gestores:
    - Cadastrar professores.
    - Redefinir senhas de professores.
    - Cadastrar categorias de documentos.
    - Enviar modelos de documentos.
    - Visualizar modelos de documentos.
    - Visualizar documentos preenchidos.
- Para Professores:
    - Visualizar e baixar modelos de documentos.
    - Enviar documentos preenchidos.

4. O usuário realiza a ação desejada e interage com o software usando uma interface gráfica na web.
5. O usuário pode sair do software a qualquer momento por meio da opção de logout, que encerra a sessão de usuário.

## 5. CASOS DE USO

Funcionalidade	Perfis de Usuário		
	Diretor	Gestor	Professor
Cadastrar gestores	✓	✗	✗
Cadastrar professores	✓	✓	✗
Redefinir senha de gestores	✓	✗	✗
Redefinir senha de professores	✓	✓	✗
Cadastrar categorias de documentos	✓	✓	✗

Enviar modelo de documentos	✓	✓	✗
Visualizar modelos de documentos	✓	✓	✓
Baixar modelos de documentos	✓	✓	✓
Enviar documentos preenchidos	✗	✗	✓
Visualizar documentos preenchidos	✓	✓	✗