



DOCUMENTAÇÃO TÉCNICA

APLICAÇÃO WEB - ADMIN

Nome do projeto:

eManager

Data de início / Prazo:

XX/XX/20223 / 6 semanas

Detalhes técnicos:

Aplicação web em React.JS

Backend em NodeJS + Adonis 5 + Lucid

Profissionais envolvidos:

Gerente de negócios: Jhollyfer Rodrigues

Gerente de projetos: Jhollyfer Rodrigues

1. REQUISITOS

1.1. REQUISITOS FUNCIONAIS:

TELA DE AUTENTICAÇÃO

[RF001] LOGIN:

Descrição: O administrador preenche os campos solicitados, em seguida aperta o botão “**entrar**”.

Entrada:

- Email*;
- Senha*;

Saída: Retornará a página inicial de importações.

Pré-condições: O usuário deve estar cadastrado no sistema e estar na tela de login.

Pós-condições: Nenhuma.

TELA DE EXTRAÇÕES

[RF002] LISTAGEM DE EXTRAÇÕES DE LEIAUTES (XMLs):

Descrição: Uma tela que irá conter todas as importações efetuadas na plataforma.

Entrada: Nenhuma.

Saída: A listagem conterá colunas com as seguintes informações:

- Nome do arquivo
- Tipo do arquivo
- Categoria
- Status
- Data

Quando o admin selecionar “**visualizar detalhes**” abrirá uma modal com as seguintes informações:

- Todos os campos com informações salvas e configuradas no tipo de coleção;

Pré-condições: o usuário deve ter feito o login e estar na tela de “Extrações”.

Pós-condições: As informações das importações ficam habilitadas para serem utilizadas em ações restritas do sistema, como também poderão ser utilizadas em telas.

[RF003] FILTRO DE EXTRAÇÕES:

Descrição: O administrador poderá efetuar um filtro de listagem de extrações (tipo, categoria, data e status).

Entrada: Tipo de filtro (tipo, categoria, data e status).

Saída: A listagem conterá colunas com as seguintes informações filtradas:

- Nome do arquivo
- Tipo do arquivo
- Categoria
- Status

Quando o admin selecionar “**filtrar**” a listagem deverá ser atualizadas com o tipo de filtro selecionado:

- Todos os campos com informações salvas e configuradas no tipo de coleção;

Pré-condições: o usuário deve ter feito o login e estar na tela de “Importações”.

Pós-condições: As informações das importações ficam habilitadas para serem utilizadas em ações restritas do sistema, como também poderão ser utilizadas em telas.

TELA DE GERENCIAMENTO DE LEIAUTES

[RF004] REGISTRO DE LEIAUTE:

Descrição: Quando o administrador seleciona a opção “Adicionar arquivo/leiaute”, é possível configurar um novo leiaute e adicionar na plataforma.

Entrada: Documento modelo XML no padrão S-2200, S-2205, S-2206, S-2230, S-2298, S-2299, S-2300, S-2306, S-2399, S-1200 e/ou S-1210 (entre outros arquivos e-social).

Saída: Janela informando que coleção foi criada

Pré-condições: O administrador deve ter feito o login no sistema, deve estar na “gerenciamento de leiaute” e ter selecionado “Adicionar arquivo/leiaute”.

Pós-condições: Leiaute criado e já disponível para extrações.

[RF005] ALTERAÇÃO DE LEIAUTE:

Descrição: Quando o administrador seleciona a opção “Alterar arquivo/leiaute”, é possível configurar o leiaute.

Entrada: Informações de leiaute a ser alterada.

Saída: Janela informando que a coleção foi alterada

Pré-condições: O administrador deve ter feito o login no sistema, deve estar na “gerenciamento de coleção” e ter selecionado “Alterar arquivo/leiaute”.

Pós-condições: Leiaute alterada e disponível para extrações.

[RF006] BLOQUEIO DE LEIAUTE:

Descrição: Quando o administrador seleciona a opção “Bloquear arquivo/leiaute”, a coleção é desativada da plataforma.

Entrada: Nenhuma.

Saída: Retorna uma janela de confirmação de bloqueio.

Pré-condições: O administrador deve ter feito o login no sistema, deve estar na “gerenciamento de leiaute” e ter selecionado “ver mais” de uma coleção específica.

Pós-condições: Leiaute desativado e não listado para Leiautes.

[RF006] LISTAGEM DE LEIAUTES:

Descrição: Uma tela de listagem de e gerenciamento das coleções configuradas na plataforma.

Entrada: Nenhuma.

Saída: A listagem conterá colunas com as seguintes informações:

- Nome do arquivo
- Tipo do arquivo
- Categoria
- Status
- Data.

Pré-condições: O administrador deve ter feito o login no sistema, deve estar na “gerenciamento de leiaute”.

Pós-condições: As informações das importações ficam habilitadas para serem utilizadas em ações restritas do sistema, como também poderão ser utilizadas em telas.

LOGOUT

[RF006] LOGOUT:

Descrição: Ao usuário pressionar sobre o botão de “**Sair**”, onde será encerrada a sua sessão no sistema, fazendo com que seja necessário uma nova sessão caso o usuário queira logar novamente na aplicação.

Processo: Nenhuma.

Entrada: Nenhuma.

Saída: Nenhuma.

Pré-condições: Estar logado no “Admin”.

Pós-condições: Nenhuma.

1.2. REQUISITOS NÃO FUNCIONAIS:

[RNF001] Log de operações do sistema:

Descrição: Possibilidade de ter um log de cada ação efetuada na plataforma.

[RNF002] Implementação

Descrição: A API deve ser desenvolvida utilizando Node.JS + Adonis 5 + Lucid

[RNF003] Implantação:

Descrição: A API e a base de dados deverá ser hospedada no Render, o Admin Web deverá ser hospedado na Vercel.

[RNF004] Usabilidade:

Descrição: A aplicação deverá ser acessível via teclado (Tab, Enter, Up, Down, Left e Right).

[RNF005] Segurança:

Descrição: O sistema deve proporcionar as seguintes seguranças:

- **[RNF005.1] Segurança do usuário:** Todas as funcionalidades que fazem parte de um perfil específico devem ser acessadas por um usuário (cadastrado) de acordo com sua função, onde sua autenticidade é verificada por token. Os dados sigilosos são criptografados.
- **[RNF005.2] Segurança do sistema:** O sistema deve sempre se manter com a proteção do certificado SSL.

[RNF006] Disponibilidade:

Descrição: O sistema não deverá operar em modo offline, havendo a necessidade de conexão com a internet. Deve permanecer disponível para os usuários 24/7 de acordo com o servidor em nuvem contratado. (desconsiderando essa hipótese caso os recursos do servidor sejam gratuitos).

[RNF007] Interoperabilidade:

Descrição: A API deve se comunicar apenas com um banco de dados de instância SQL (PostgreSQL ou MySQL).

Obs: Relatórios de alta especificidade contemplam apenas o banco de produção.

[RNF008] Portabilidade:

Descrição: A aplicação deve ser executada em total funcionalidade nos navegadores Google Chrome, Mozilla Firefox, Safari em versão atual do dia de início do projeto.

[RNF009] Ético:

Descrição: O sistema não irá (E não deverá) expor dados privativos para qualquer pessoa ou organização que não seja proprietária do mesmo (exceto por meios judiciais).

[RNF010] Legais:

Descrição: O sistema deve proporcionar os seguintes critérios legais:

- **[RNF010.1] Política de privacidade:** O sistema deve respeitar as políticas de privacidade impostas.
- **[RNF010.2] Lei LGPD:** O sistema deve respeitar a nova lei LGPD (Lei Geral de Proteção de Dados) que assegura aos usuários de solicitar a qualquer momento a exclusão de seus dados por definitivo é também possível obter todas as informações de cunho pessoal.

[RNF011] Entrega:

Descrição: O cronograma deve ser atualizado com as novas entregas a cada duas semanas dando a visão do deadline do projeto.

2. PRODUTOS E PRAZOS

2.1. HOSPEDAGENS:

2.1.1. Aplicação web: Vercel (Hobby)

Sites não comerciais, sem requisitos comerciais (gratuito):

- CI/CD integrado
- SSL e HTTPS automáticos
- Pré-visualizações do Git Push

2.1.2. API: Render (Starter)

Serviços não comerciais, sem requisitos comerciais:

- CI/CD integrado
- SSL e HTTPS automáticos
- Pré-visualizações do Git Push
- RAM: 512MB
- CPU: 0.1
- Preço: \$7/mês

2.1.2. Base de dados: Render (Starter)

Serviços não comerciais, sem requisitos comerciais:

- CI/CD integrado
- SSL e HTTPS automáticos
- Pré-visualizações do Git Push
- RAM: 512MB
- CPU: 0.1
- Armazenamento: 1BG
- Preço: \$7/mês

Durante o desenvolvimento os ambientes irão utilizar os planos gratuitos:

3.2. Prazos e entregas:

3.2.1. Features

A cada semana deverá ser entregue um relatório com as features desenvolvidas com:

- Descrição
- Status
- Links testáveis

3.2.2. Cronograma

Fornecer um cronograma com os status das features

4. ESPECIFICAÇÕES TÉCNICAS

4.1. FRAMEWORKS E BIBLIOTECAS:

4.2.1. Aplicação web: React.JS - Material UI.

4.2.2. Backend: Node.JS - Adonis 5 - Lucid

4.2. REPOSITÓRIOS:

4.3.1. Backend: Github:

4.3.2. Aplicação web: Githubl

4.3. DETALHES:

4.4.1. TECNOLOGIAS:

4.4.1.1. Backend: Typescript, NodeJS, Adonis 5, Lucid

4.4.1.2. Aplicação web: Typescript, React.JS, Material UI