**Khosta : Sistema de Gestão Penitenciária**

# ****Introdução****

## ****Contextualização do Tema****

O sistema penitenciário enfrenta diversos desafios na gestão de reclusos, visitantes e registos, sobretudo quando os processos são manuais ou pouco eficazes. A falta de organização e transparência pode levar a erros administrativos, fugas, visitas não autorizadas e dificuldades na emissão de relatórios.

A informatização da gestão penal surge como uma solução para melhorar o controlo de dados, reforçar a segurança e agilizar processos burocráticos. O **Khosta Penal** propõe-se a ser um sistema moderno, intuitivo e seguro, capaz de centralizar todas as operações essenciais de uma instituição prisional.

## ****Problematização****

Actualmente, muitos estabelecimentos prisionais ainda dependem de registos manuais ou sistemas desactualizados, o que gera diversos problemas, tais como:

* **Falta de controlo eficaz** sobre o movimento de reclusos e visitantes.
* **Dificuldade na emissão de relatórios** actualizados e fiáveis.
* **Riscos de segurança** devido a permissões mal definidas ou acesso indevido a informações.
* **Processos morosos e sujeitos a erros**, afectando a produtividade dos agentes prisionais.

### ****1.3 Justificação****

A implementação do **Khosta Penal** justifica-se pela necessidade de modernizar a gestão prisional, trazendo benefícios como:

* **Maior transparência** nos registos de reclusos e visitas.
* **Redução de erros humanos** através da automatização de processos.
* **Segurança reforçada** com permissões de acesso baseadas em perfis de utilizador.
* **Emissão rápida de relatórios** para análise e tomada de decisões.

## ****Objectivos****

### ****Objectivo Geral****

Desenvolver um sistema de gestão penal (**Khosta**) que permita o controlo eficaz de reclusos, visitantes e relatórios, com permissões de acesso diferenciadas consoante o perfil do utilizador.

### ****Objectivos Específicos****

* **Criar uma interface funcional** para registo e consulta de reclusos.
* **Implementar um sistema de permissões** baseado em tipos de utilizador (administrador, agente prisional, director).
* **Permitir a gestão de visitantes** e a emissão de relatórios automáticos.
* **Utilizar tecnologias web modernas**, como **HTML, CSS, JavaScript, Tailwind CSS e LocalStorage**, para garantir uma experiência responsiva e eficiente.

# Descrição do Projeto

## Visão Geral

O **Khosta Penal** é um sistema web desenvolvido com o propósito de otimizar a gestão administrativa de estabelecimentos prisionais. O projeto visa digitalizar e centralizar informações cruciais, como o registo de reclusos, controle de visitas e emissão de relatórios, proporcionando um ambiente mais seguro, transparente e eficiente. O sistema foi projetado para ser intuitivo e responsivo, utilizando tecnologias web modernas (HTML, CSS, JavaScript, Tailwind CSS) e armazenamento local (LocalStorage) para persistência de dados em ambiente local.

A plataforma permite que diferentes perfis de utilizador (administrador, agente prisional, diretor) acedam a funcionalidades específicas conforme as suas permissões, reforçando a segurança e a organização dos processos internos da instituição prisional.

## Estrutura do Sistema

O **Khosta Penal** é composto por um conjunto de páginas que representam as principais funcionalidades da aplicação. A estrutura modular facilita a navegação e a manutenção do sistema, além de permitir futuras expansões. Abaixo, descrevem-se as páginas atualmente implementadas (ou previstas):

* index.html **– Tela de Login**
* dashboard.html **– Painel Principal**
* cadastro.html **– Cadastro de Reclusos**
* lista.html **– Listagem e Busca de Reclusos**
* relatorios.html **– Relatórios Administrativos**
* visitantes.html **– Cadastro de Visitantes (em desenvolvimento)**

## **Deploy na Nuvem**

Para facilitar o acesso remoto e os testes do sistema, o Khosta Penal foi publicado na plataforma **Vercel**, que oferece deploy contínuo para aplicações web estáticas ou baseadas em frontend moderno.

O deploy permite que o sistema seja acessado via navegador, sem necessidade de instalação local. Isso facilita a demonstração, testes em diferentes dispositivos e o uso por múltiplos usuários simultaneamente (em ambiente de demonstração).

# ****Requisitos do Sistema****

A definição dos requisitos é fundamental para garantir que o sistema **Khosta** atenda às necessidades funcionais e operacionais do ambiente prisional. Esta seção apresenta os requisitos funcionais e não funcionais identificados durante o levantamento e planejamento do projeto.

## Requisitos Funcionais

Os requisitos funcionais representam as **funcionalidades essenciais** que o sistema deve oferecer para cumprir seus objetivos. Eles descrevem comportamentos esperados e interações entre os utilizadores e o sistema:

* **RF01: Login de usuário com tipo (admin, agente, diretor)**  
  O sistema deve permitir que diferentes perfis de utilizador realizem login. Após a autenticação, o utilizador será redirecionado ao painel correspondente ao seu nível de acesso (administrador, agente prisional ou diretor).
* **RF02: Cadastro e edição de prisioneiros**  
  Deve ser possível inserir novos registos de reclusos, com dados como nome, identificação, crime, data de entrada, tempo de pena, entre outros. Também é necessário permitir a edição dessas informações conforme necessário.
* **RF03: Listagem e filtro de prisioneiros**  
  O sistema deve listar todos os prisioneiros cadastrados, com possibilidade de aplicar filtros por nome, tipo de crime, situação penal, entre outros critérios relevantes.
* **RF04: Geração de relatórios por situação, crime, etc.**  
  O sistema deve gerar relatórios com base em parâmetros selecionados, como tipo de crime, situação (preventiva, condenatória, em liberdade condicional), tempo de pena, entre outros, para fins de análise e tomada de decisão.
* **RF05: Controle de visitantes**   
  Deve ser possível, futuramente, registrar e controlar o acesso de visitantes, com dados como nome, parentesco, datas e horários de visita, e prisioneiro relacionado.
* **RF06: Interface de administração de usuários (admin)**  
  O Administrador deve poder cadastrar, editar ou remover outros utilizadores do sistema, definindo seus respectivos perfis de acesso.
* **RF07: Controle de permissões por perfil**  
  Cada tipo de utilizador terá acesso apenas às funcionalidades autorizadas. Por exemplo, agentes não poderão editar perfis de utilizadores, e diretores poderão acessar relatórios estratégicos.

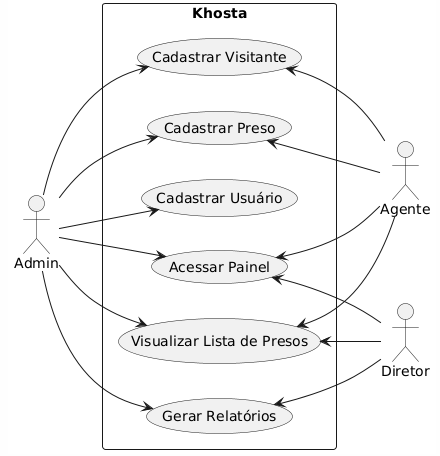
## Requisitos Não Funcionais

Os requisitos não funcionais determinam **características de qualidade** do sistema, como desempenho, usabilidade, arquitetura e restrições tecnológicas. No contexto do **Khosta Penal**, destacam-se:

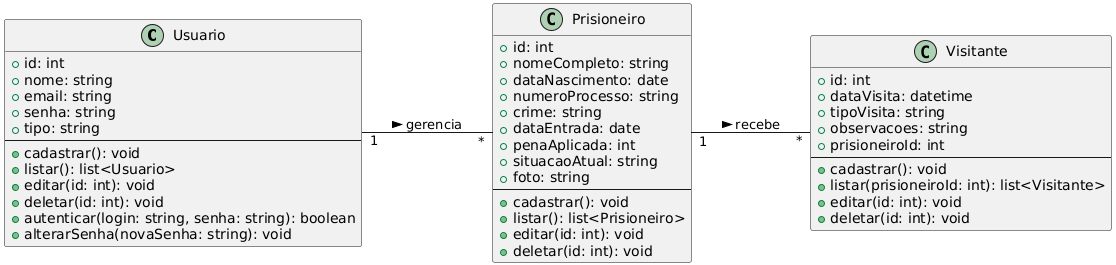
* **RNF01: Interface responsiva (TailwindCSS)**  
  A interface do sistema deve adaptar-se a diferentes tamanhos de tela (desktop, tablet, mobile), garantindo uma boa experiência de utilização em qualquer dispositivo.
* **RNF02: Persistência local via LocalStorage**  
  Os dados devem ser armazenados localmente no navegador, utilizando a API LocalStorage, possibilitando o uso offline em ambientes restritos, sem necessidade de conexão com servidor externo.
* **RNF03: Usabilidade e feedback visual com SweetAlert2**  
  O sistema deve utilizar feedbacks visuais claros (como notificações de sucesso, erro e confirmação) através da biblioteca SweetAlert2, proporcionando uma interação mais amigável e compreensível.
* **RNF04: Código modularizado em arquivos JS**  
  O código JavaScript deve ser organizado de forma modular, com separação de responsabilidades por arquivos, facilitando a manutenção, reutilização e expansão do sistema.

# ****Diagramas de Modelagem****

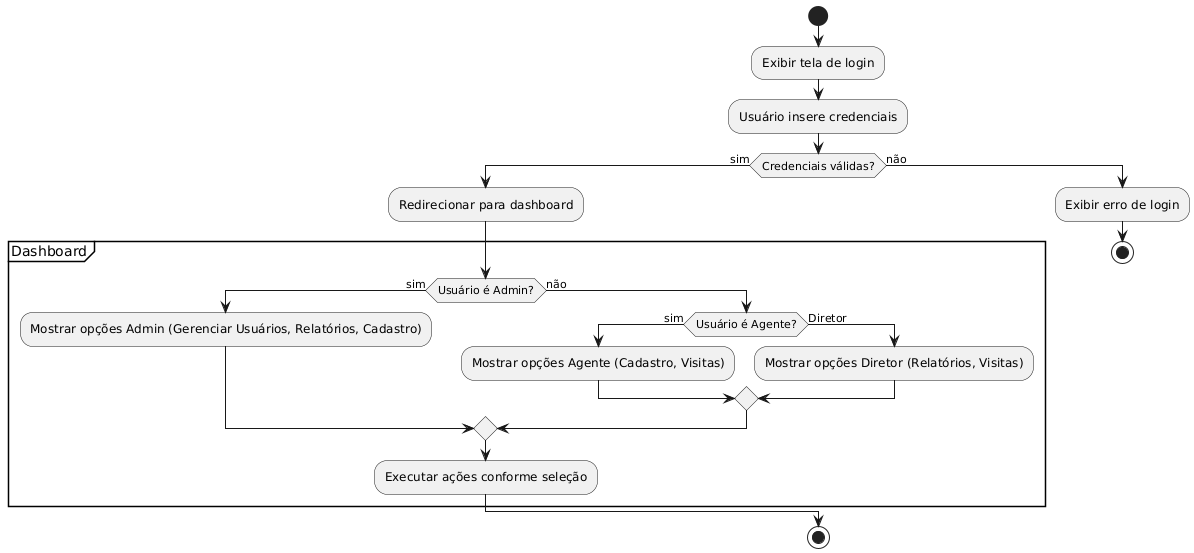
* **4.1 Diagrama de Casos de Uso**  
  Inclui atores: Admin, Agente, Diretor; casos como "Cadastrar Preso", "Gerar Relatórios", "Gerenciar Usuários", "Cadastrar Visitante".



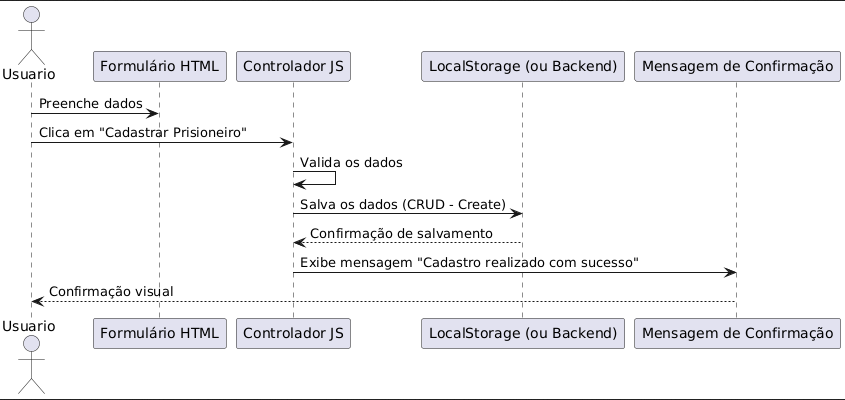
* **4.2 Diagrama de Classes**   
  Mostrar entidades: Usuario, Prisioneiro, Visitante, com atributos e métodos.



* **4.3 Diagrama de Actividades**  
  Mostrar fluxo: login → dashboard → ações conforme permissões.



**Diagrama de Sequência**  
Mostra o fluxo de s**equência** com interações entre: Usuário, Formulário HTML, Controlador JS, LocalStorage (ou Backend) e Mensagem de Confirmação. O usuário preenche os dados, o sistema valida, salva e confirma o cadastro com uma mensagem de sucesso.



### 5 Tecnologias Utilizadas

O desenvolvimento do sistema **Khosta Penal** fundamenta-se em tecnologias web acessíveis e modernas, que foram escolhidas para garantir um sistema leve, responsivo e de fácil manutenção. A seguir, descrevem-se as principais tecnologias utilizadas no projeto e suas respectivas finalidades.

#### 5.1 HTML5

Foi utilizado para estruturar semanticamente todas as páginas do sistema. Sua aplicação garante organização, acessibilidade e compatibilidade entre diferentes navegadores, além de facilitar futuras manutenções no código.

#### 5.2 CSS3 e TailwindCSS

A estilização da interface foi feita com CSS3 em conjunto com o framework TailwindCSS. Este último permite a criação rápida de interfaces modernas e responsivas por meio de classes utilitárias, agilizando o processo de desenvolvimento visual e mantendo a consistência do design.

#### 5.3 JavaScript (Puro)

A linguagem JavaScript foi empregada para controlar a lógica de interação do sistema no lado do cliente. Suas funcionalidades incluem validação de formulários, manipulação do DOM e resposta a eventos de interface, proporcionando maior dinamismo à aplicação.

#### 5.4 LocalStorage

O armazenamento de dados foi implementado com o uso da API LocalStorage, permitindo persistência local no navegador sem necessidade de conexão com banco de dados. Essa abordagem simplifica a estrutura inicial e garante funcionamento mesmo em ambientes offline.

#### 5.5 SweetAlert2

Utilizada para melhorar a comunicação com o usuário, a biblioteca SweetAlert2 permite exibir mensagens personalizadas de sucesso, erro ou confirmação. Com isso, a experiência visual torna-se mais intuitiva e agradável.

#### 5.7 Vercel

A plataforma Vercel foi escolhida para hospedagem e deploy contínuo do sistema. Além de gratuita, ela permite atualizações automáticas a partir do repositório, tornando a aplicação disponível online de forma prática e eficiente para testes, demonstrações e uso em produção.

## 6. Conclusão

\*\*Conclusão Humanizada\*\*

O desenvolvimento do \*\*Khosta Penal\*\* representou mais do que um projeto técnico—foi uma oportunidade de contribuir para a modernização e humanização da gestão prisional, um setor que muitas vezes enfrenta desafios complexos, desde a burocracia até a falta de transparência. Ao longo desta primeira fase, buscamos não apenas cumprir requisitos funcionais, mas também criar uma ferramenta que pudesse, de fato, facilitar o trabalho dos agentes penitenciários, diretores e administradores, tornando processos manuais e desgastantes em tarefas ágeis e seguras.

\*\*Impacto e Valor Humano\*\*

Acreditamos que a tecnologia, quando bem aplicada, pode transformar realidades. O Khosta Penal não é apenas um sistema de registros—é uma ponte para maior organização, redução de erros e, principalmente, para um ambiente mais controlado e justo, onde informações sobre reclusos e visitantes são acessíveis de forma clara e responsável. Sabemos que, em um contexto prisional, cada minuto poupado em planilhas ou buscas manuais pode ser redirecionado para ações mais significativas, como a reintegração ou a segurança de todos dentro do sistema.

\*\*Desafios e Aprendizados\*\*

Trabalhar com armazenamento local (\*LocalStorage\*) e tecnologias front-end foi um passo inicial importante para validar a ideia, mas também nos lembrou das limitações que precisam ser superadas. Ouvir feedbacks de potenciais usuários—agentes, diretores—nos mostrou que a jornada não termina aqui. A migração para um \*backend\* robusto, a implementação de autenticação mais segura e a expansão de funcionalidades (como o módulo de visitantes) são sonhos que carregamos para as próximas fases, sempre com o foco em \*\*servir melhor\*\* quem depende deste sistema no dia a dia.

\*\*Gratidão e Futuro\*\*

Este projeto não seria possível sem a base de conhecimento compartilhado por tantas fontes abertas e comunidades de desenvolvimento. Cada linha de código foi escrita com a esperança de que, em breve, o Khosta Penal possa sair do ambiente de testes e se tornar uma ferramenta real, ajudando a descomplicar a gestão penitenciária em nosso contexto.

Por fim, entendemos que a tecnologia é uma ferramenta—e o que importa é como ela melhora vidas. Seguiremos trabalhando com essa missão.

\*\*Equipe Khosta Penal\*\*

\*"Tecnologia a serviço de uma gestão mais humana."\*

---

\*(Nota: Se desejar, podemos incluir um depoimento fictício ou real de um "usuário-teste" para dar ainda mais emotividade, como: "Antes, perdíamos horas buscando fichas. Agora, com o Khosta, tudo está a um clique." Exemplo adaptável ao seu público.)\*

## 7. Referências Bibliográficas

Mozilla Foundation. (2023). HTML: HyperText Markup Language. MDN Web Docs. <https://developer.mozilla.org/en-US/docs/Web/HTML>

Mozilla Foundation. (2023). CSS: Cascading Style Sheets. MDN Web Docs. <https://developer.mozilla.org/en-US/docs/Web/CSS>

Mozilla Foundation. (2023). JavaScript. MDN Web Docs. <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

Tailwind Labs. (2024). Tailwind CSS Documentation. <https://tailwindcss.com/docs>

SweetAlert2. (2024). SweetAlert2: Beautiful, responsive, customizable replacement for JavaScript’s popup boxes. <https://sweetalert2.github.io/>

W3C. (2023). Web Storage (LocalStorage) API. <https://www.w3.org/TR/webstorage/>

Vercel. (2024). Vercel Documentation. <https://vercel.com/docs>

Pressman, R. S., & Maxim, B. R. (2016). Engenharia de Software (8ª ed.). McGraw-Hill.

Nielsen, J. (1994). Usability engineering. Morgan Kaufmann.

# Apêndice A – Interfaces do Sistema

Este apêndice apresenta as principais interfaces gráficas do sistema **Khosta Penal**, com descrições das funcionalidades disponíveis em cada tela. A exibição e disponibilidade das funcionalidades variam de acordo com o tipo de utilizador (administrador, agente prisional ou diretor).

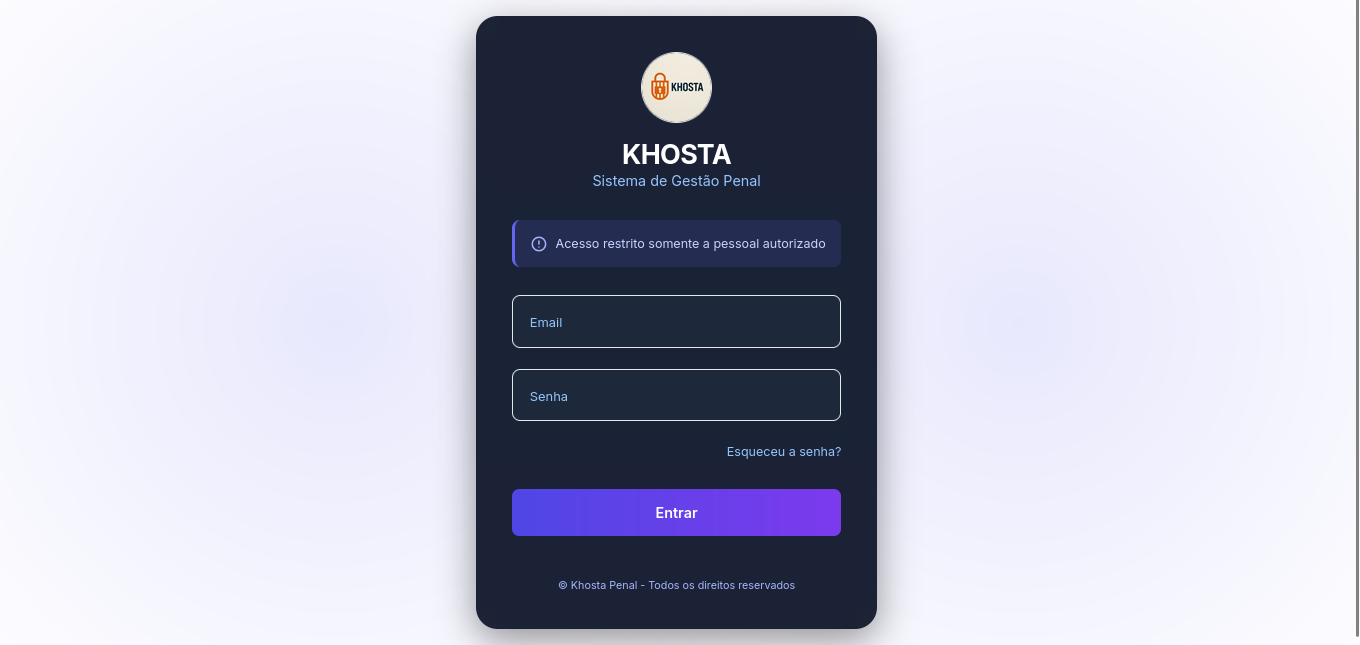
**Nota:** Prints de tela podem ser inseridos abaixo de cada descrição, conforme as capturas forem obtidas.

## A.1 Tela de Login (index.html)

**Descrição:**  
Tela inicial do sistema, onde o utilizador informa suas credenciais para acesso. Após login bem-sucedido, o utilizador é redirecionado ao painel correspondente ao seu perfil.

**Funcionalidades:**

* Campo de nome de utilizador.
* Seleção do tipo de utilizador (admin, agente, diretor).
* Botão de login com validação.
* Alertas visuais em caso de erro.

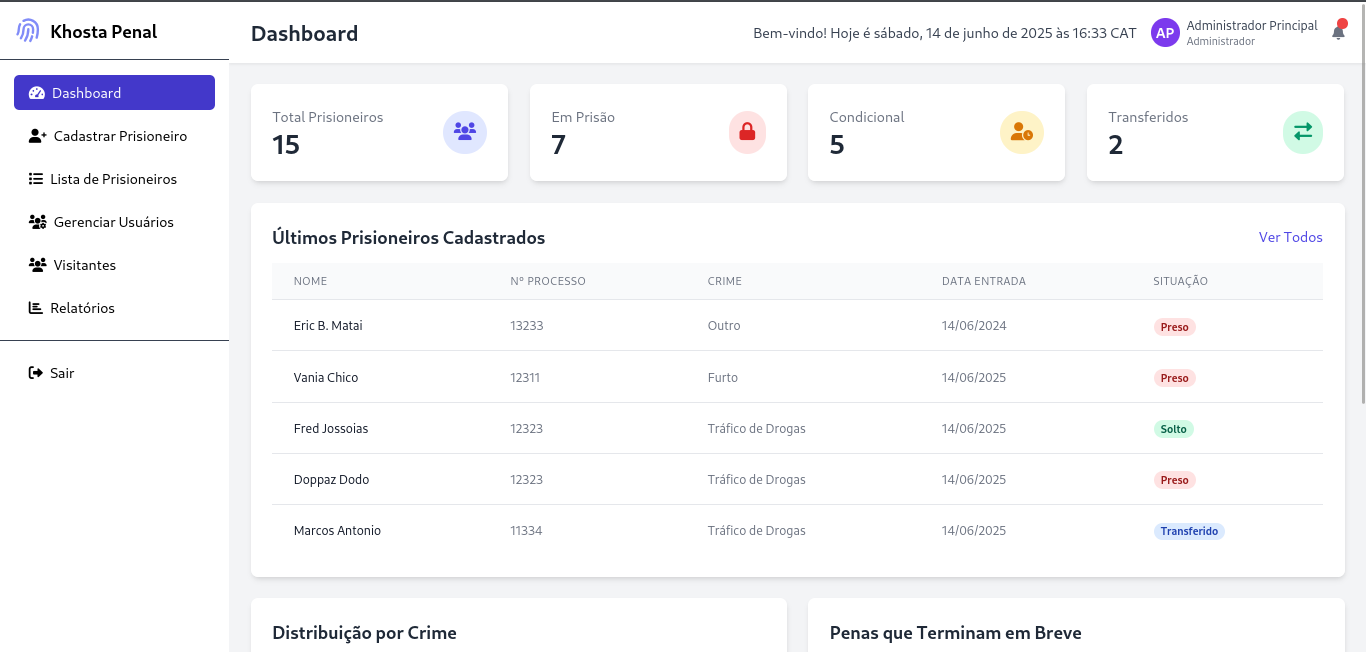


## A.2 Dashboard (dashboard.html)

**Descrição:**  
Painel central que serve como ponto de partida após o login. Exibe atalhos para as funções disponíveis de acordo com o perfil de utilizador.

**Funcionalidades:**

* Acesso rápido a módulos como cadastro, relatórios, visitantes, etc.
* Visualização de dados resumidos (número de prisioneiros, por exemplo).
* Ícones ou botões estilizados com TailwindCSS para navegação.

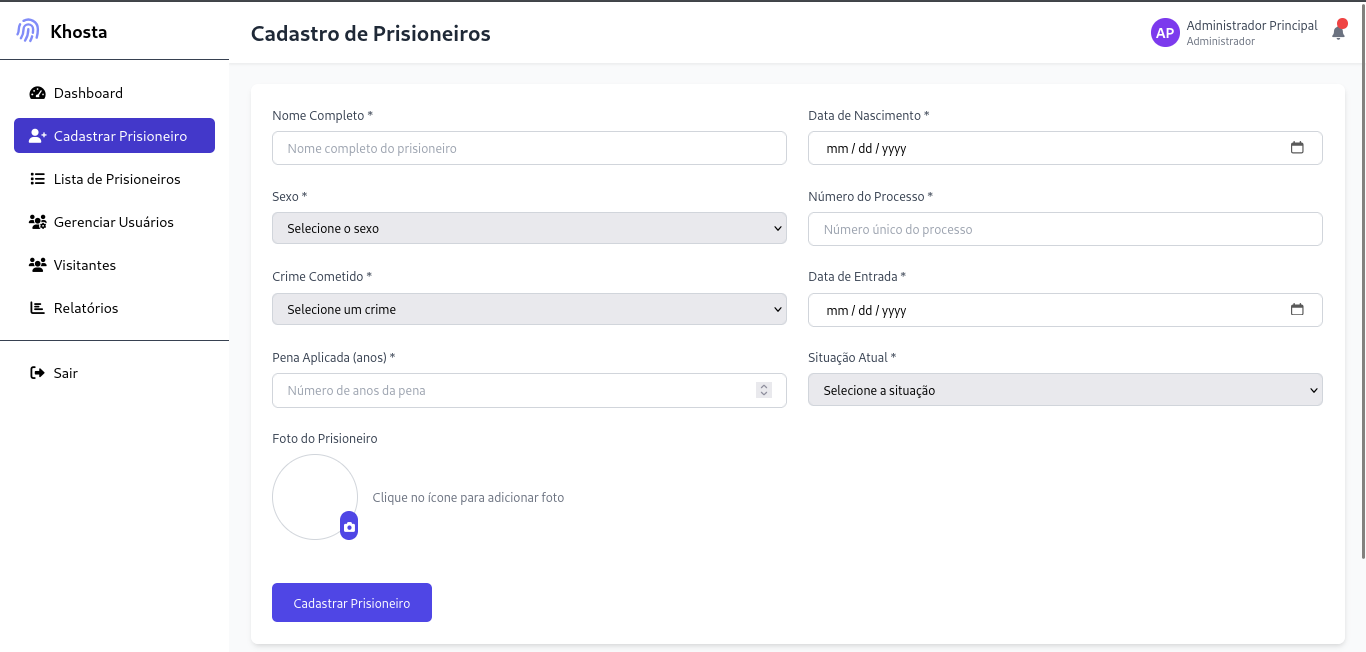


## A.3 Cadastro de Prisioneiro (cadastro.html)

**Descrição:**  
Tela para registrar novos prisioneiros ou editar registros existentes.

**Funcionalidades:**

* Formulário com campos como nome, crime, tempo de pena, etc.
* Botões para salvar, cancelar e limpar formulário.
* Feedback visual após operações com SweetAlert2.

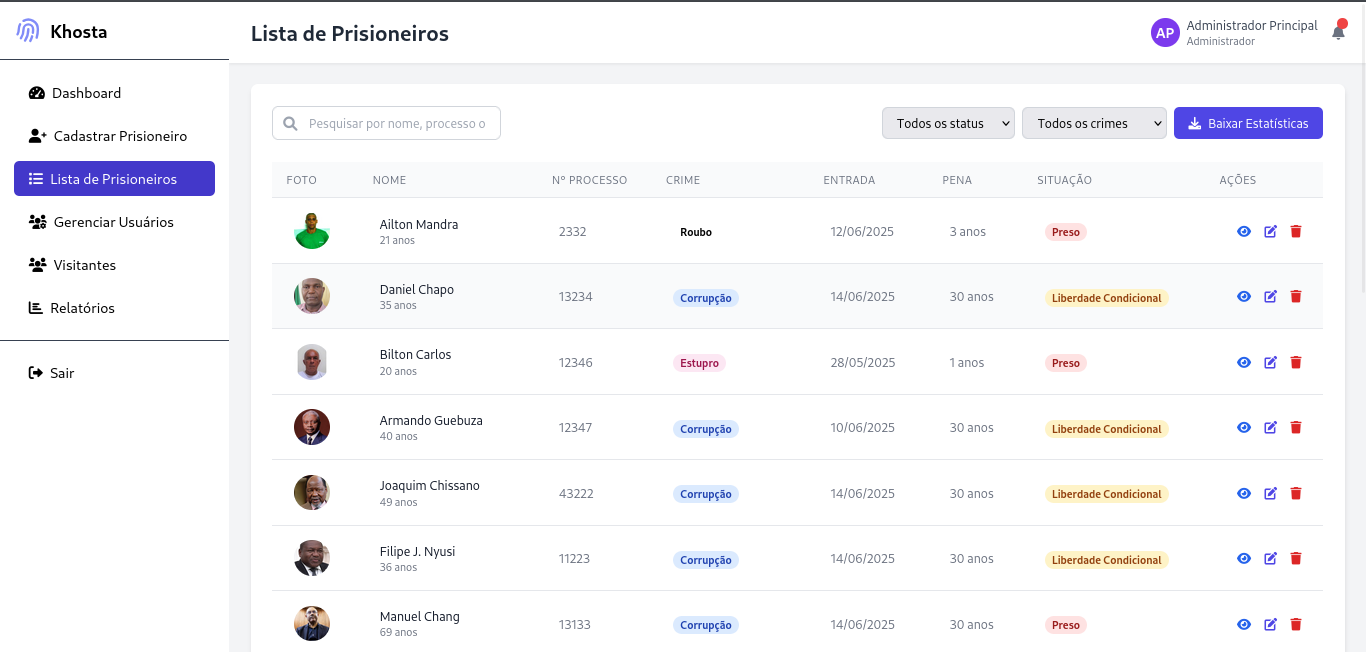


## A.4 Lista de Prisioneiros (lista.html)

**Descrição:**  
Exibição de todos os prisioneiros cadastrados no sistema, com filtros e ações específicas.

**Funcionalidades:**

* Tabela interativa com dados principais.
* Filtros por nome, crime, situação, etc.
* Botões para editar ou excluir (conforme permissões do utilizador).

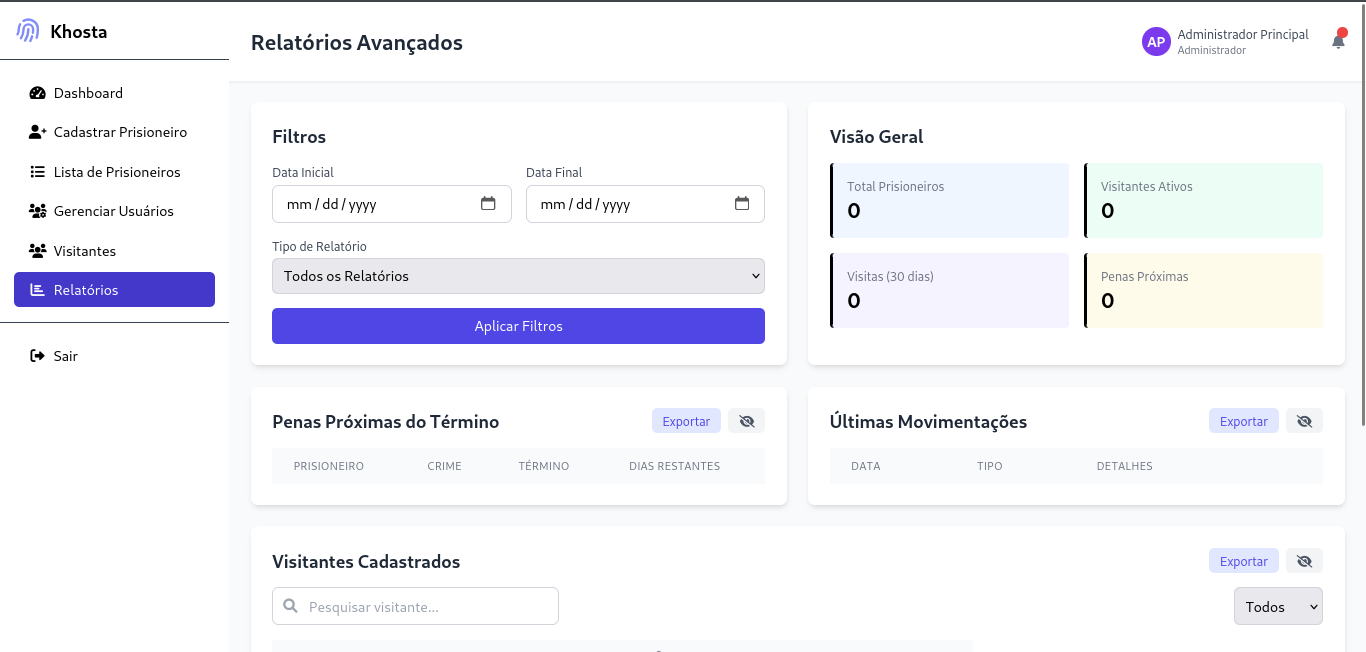


## A.5 Relatórios (relatorios.html)

**Descrição:**  
Geração de relatórios administrativos e estatísticos, destinados a diretores e gestores.

**Funcionalidades:**

* Filtros avançados (tipo de crime, tempo de pena, status do prisioneiro).
* Apresentação dos resultados em listas ou gráficos (quando implementado).
* Exportação de dados (futura funcionalidade opcional).

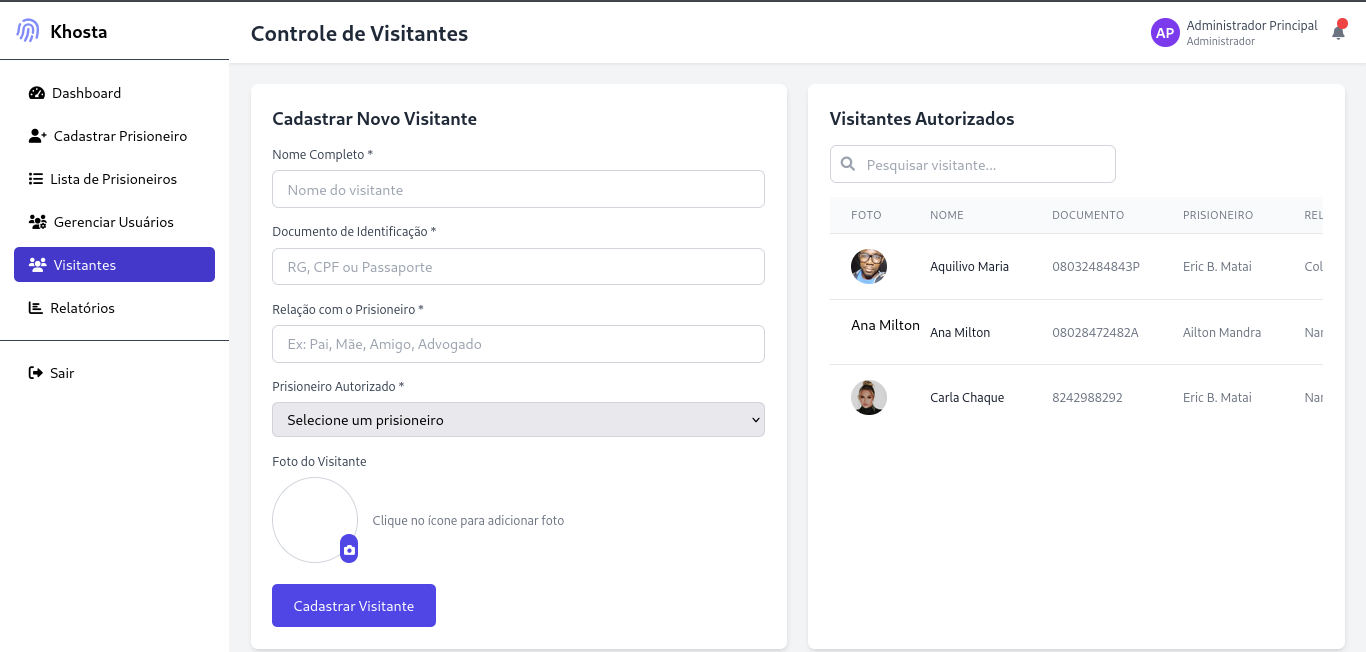


## A.6 Controle de Visitantes (visitantes.html) (Planejada)

**Descrição:**  
Interface para controle de visitas aos reclusos. Esta funcionalidade encontra-se em desenvolvimento.

**Funcionalidades previstas:**

* Cadastro de visitantes com vínculo ao prisioneiro.
* Registro de data e horário da visita.
* Histórico de visitas por prisioneiro.



## A.7 Gerenciamento de Usuários (Admin)

**Descrição:**  
Área exclusiva para administradores realizarem a gestão de contas de utilizador do sistema.

**Funcionalidades:**

* Cadastro de novos utilizadores.
* Edição e exclusão de utilizadores existentes.
* Definição de permissões conforme o tipo de utilizador.

