

Componentes Básicos

A aplicação Pharmabot foi construída utilizando o Next.js, um framework moderno e robusto baseado em React, amplamente adotado para o desenvolvimento de aplicações web escaláveis, performáticas e com uma estrutura organizacional clara. O Next.js oferece uma série de funcionalidades que o tornam ideal para projetos que exigem modularidade, rápida iteração e facilidade de manutenção, como é o caso da interface do Pharmabot.

Figura 1 - Next.js*



Entre os principais benefícios do Next.js, destaca-se o suporte nativo à componentização, o que significa que cada elemento da interface pode ser desenvolvido como uma unidade isolada de código, com responsabilidades bem definidas. Essa abordagem não apenas melhora a legibilidade e organização do projeto, como também permite uma maior reutilização de componentes, testabilidade individual e independência entre partes da interface. Isso foi essencial para o desenvolvimento do Pharmabot, que exige múltiplas visualizações reutilizando a mesma estrutura de dados — como tabelas, botões e modais — com comportamentos distintos.

Ainda, como já citado, no front-end da Pharmabot está o React, uma biblioteca declarativa e baseada em componentes, responsável por controlar o estado e o comportamento da interface do usuário. O React é amplamente reconhecido por sua abordagem reativa à construção de interfaces, permitindo que a UI se atualize automaticamente em resposta a mudanças de estado — o que se encaixa perfeitamente com os requisitos de aplicações como o Pharmabot, que demandam feedbacks imediatos ao usuário, eventos dinâmicos (como cliques em botões de parada ou seleção de bins) e atualização contínua de dados na tela.

A escolha do React também foi motivada pela sua capacidade de criar componentes reutilizáveis, isolados e declarativos, que encapsulam lógica e estilo em unidades coesas e reutilizáveis. Isso permitiu a criação de componentes como TabelaPharma, FormModal e StopButton com comportamentos bem definidos e reutilização em múltiplas partes da aplicação, reduzindo redundância e facilitando manutenção.

Figura 2 - React.js



Para a camada visual da aplicação, optamos por utilizar o MUI (Material UI), uma das bibliotecas de componentes visuais mais completas e populares do ecossistema React. Baseado nas diretrizes de design do Material Design da Google, o MUI oferece uma vasta gama de componentes estilizados prontos para uso — como botões, inputs, selects, tabelas e modais — todos com acessibilidade, responsividade e usabilidade embutidas.

Alguns pontos foram essenciais para a escolha do MUI no projeto, como a agilidade no desenvolvimento, padronização visual e a integração nativa com o react. Sendo assim, na prática, diversos elementos da interface do Pharmabot foram construídos ou aprimorados com o uso do MUI. Entre os principais componentes MUI utilizados no projeto, destacam-se:

- Box: usado extensivamente como componente de estruturação, oferecendo flexibilidade para construir containers e organizar elementos com paddings, margins e layouts responsivos.
- Select, MenuItem, InputLabel e FormControl: empregados na construção de menus suspensos de seleção, como na definição de parâmetros de operação e filtragem de dados. Esses componentes oferecem suporte completo a acessibilidade, foco automático e personalização visual.

Figura 3 - MUI - Bibliotecas



- SwipeableDrawer: utilizado para construir a Sidebar responsiva da aplicação, com suporte a gestos de toque e abertura por swipe, especialmente útil para dispositivos móveis ou interações fluídas com o usuário.
- IconButton e MenuIcon: empregados para construir o botão hamburguer que controla a abertura da sidebar, com ícones intuitivos e animações suaves.
- List, ListItem, ListItemButton, ListItemText e Divider: responsáveis pela estrutura da navegação lateral, organizando as rotas da aplicação com uma hierarquia clara, separação visual e interações consistentes.
- Container: utilizado para centralizar e estruturar o conteúdo principal da tela, garantindo uma boa experiência visual tanto em telas amplas quanto em dispositivos menores.

Todos esses componentes foram integrados de forma modular nos arquivos da aplicação, com destaque para componentes como Sidebar.tsx, SelectButton.tsx, FormModal.tsx, StopButton.tsx, Header.tsx e TabelaPharma.tsx. O uso do MUI também permitiu que a interface mantivesse uma consistência visual e de comportamento, alinhada com boas práticas de UX.

Entre os componentes básicos mais utilizados no projeto, destacam-se:

O componente Header.tsx é responsável por exibir o cabeçalho fixo da aplicação, presente em todas as telas principais. Ele contém o título da aplicação, mantendo a identidade visual do sistema, e pode incluir elementos como botões de ação, ícones ou menus contextuais, dependendo da tela. Utiliza o componente Box do MUI para estruturação do layout, garantindo alinhamento, espaçamento e responsividade. Sua função principal é fornecer uma referência visual constante ao usuário, reforçando a hierarquia da interface e melhorando a navegabilidade.

Figura 4 - Header das telas do sistema



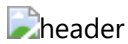
A Sidebar.tsx é o componente responsável pela navegação lateral da aplicação. Construída com o componente SwipeableDrawer do MUI, ela oferece uma experiência fluida e responsiva, com suporte a gestos de arrastar em dispositivos móveis. A sidebar contém uma lista de rotas navegáveis, organizadas com os componentes List, ListItem e ListItemButton, além de um botão hamburguer (IconButton com MenuIcon) que permite abrir e fechar o menu. Seu uso garante uma navegação intuitiva, sem recarregamento de página, utilizando o componente Link do Next.js para roteamento interno.

Figura 5 - Sidebar das telas do sistema



O componente `FormModal.tsx` é utilizado para exibir formulários dentro de janelas modais, permitindo a criação ou edição de registros de forma não intrusiva, sem redirecionar o usuário para outra página. Baseado nos componentes do MUI como `Modal`, `TextField`, `Select`, `InputLabel` e `Button`, ele garante uma experiência fluida, acessível e visualmente consistente com o restante da aplicação. Seu conteúdo é exibido dinamicamente com base nas props recebidas, tornando-o reutilizável em diferentes contextos. O `FormModal` é uma peça central na interface do Pharmabot, sendo usado, por exemplo, na inserção de dados de medicamentos ou configurações operacionais.

Figura 6 - Modal para abrir um formulário



Agora, uns dos componentes mais utilizados a nossa `tabelaPharma`:

O componente `TabelaPharma.tsx` é uma tabela reutilizável que centraliza a exibição de dados em diferentes telas da aplicação. Ele combina o layout responsivo do `Container` do MUI com o componente `TituloTabela` para exibir título e subtítulo, além de um controle de carregamento via `CircularProgress`. Ao receber dados, colunas, configurações de paginação e funções de ação como props, o componente permite flexibilidade e padronização. Sua estrutura modular garante consistência visual e facilita a manutenção do front-end do Pharmabot.

Figura 7 - Tabela de histórico de prescrições



Para mais detalhes sobre os demais componentes desenvolvidos, bem como sua aplicação nas diferentes telas do sistema, recomenda-se a consulta direta à pasta do projeto. A estrutura completa dos componentes pode ser acessada no seguinte caminho:

```
/src/frontend/pharmabot/components
```

Nessa pasta, é possível verificar a implementação de cada componente, suas dependências, estilos aplicados e como eles se integram à interface do Pharmabot. A divisão modular facilita o entendimento do papel de cada parte da aplicação e permite uma manutenção mais eficiente ao longo do desenvolvimento.