Barra de Aplicativos

A barra de aplicativos exibe informações e ações relacionadas à tela atual.

A barra de aplicativos superior fornece conteúdo e ações relacionadas à tela atual. Ela é utilizada para a identidade visual, títulos de tela, navegação e ações.

Ela pode se transformar em uma barra de ações contextual ou ser utilizada como uma barra de navegação.

{{"component": "modules/components/ComponentLinkHeader.js"}}

Barra de Aplicativos Simples

{{"demo": "ButtonAppBar.js", "bg": true}}

Barra do site com menu

{{"demo": "MenuAppBar.js", "bg": true}}

Barra do site com menu responsivo

{{"demo": "ResponsiveAppBar.js", "bg": true}}

Barra de Aplicativos com campo de busca

Uma barra de pesquisa lateral

{{"demo": "SearchAppBar.js", "bg": true}}

Barra do site com um campo de busca principal

Uma barra de pesquisa primária

{{"demo": "PrimarySearchAppBar.js", "bg": true}}

Densa (apenas para desktop)

{{"demo": "DenseAppBar.js", "bg": true}}

Proeminente

A prominent app bar.

{{"demo": "ProminentAppBar.js", "bg": true}}

Bottom App Bar

{{"demo": "BottomAppBar.js", "iframe": true, "maxWidth": 400}}

Fixed placement

When you render the app bar position fixed, the dimension of the element doesn't impact the rest of the page. This can cause some part of your content to be invisible, behind the app bar. Here are 3 possible solutions:

1. Você pode usar `position="sticky"` ao invés de fixed. ⚠ sticky não é suportado pelo IE11.
2. Você pode renderizar um segundo componente `<Toolbar />`:

```
function App() {
  return (
    <React.Fragment>
      <AppBar position="fixed">
        <Toolbar>{/* content */}</Toolbar>
      </AppBar>
      <Toolbar />
    </React.Fragment>
  );
}
```

3. Você pode usar o CSS `theme.mixins.toolbar`:

```
const Offset = styled('div')(({ theme }) => theme.mixins.toolbar);

function App() {
  return (
    <React.Fragment>
      <AppBar position="fixed">
        <Toolbar>{/* content */}</Toolbar>
      </AppBar>
      <Offset />
    </React.Fragment>
  );
}
```

Scrolling

You can use the `useScrollTrigger()` hook to respond to user scroll actions.

Barra de aplicativos oculta

The app bar hides on scroll down to leave more space for reading.

{{"demo": "HideAppBar.js", "iframe": true}}

Barra de aplicativos elevada

The app bar elevates on scroll to communicate that the user is not at the top of the page.

{{"demo": "ElevateAppBar.js", "iframe": true}}

Voltar ao topo

A floating action buttons appears on scroll to make it easy to get back to the top of the page.

{{"demo": "BackToTop.js", "iframe": true}}

`useScrollTrigger([options]) => trigger`

Argumentos

1. `options` (*object* [opcional]):
   - `options.disableHysteresis` (*bool* [opcional]): Padrão `false`. Desabilita a histerese. Ignora a direção de rolagem ao determinar o valor de `trigger`.
   - `options.target` (*Node* [opcional]): Padrão `window`.
   - `options.threshold` (*number* [opcional]): Padrão `100`. Modifica o valor limite que aciona a `trigger` quando a barra de rolagem vertical cruzar ou chegar a este limite.

Retornos

`trigger`: Does the scroll position match the criteria?

Exemplos

```
import useScrollTrigger from '@mui/material/useScrollTrigger';

function HideOnScroll(props) {
  const trigger = useScrollTrigger();
  return (
    <Slide in={!trigger}>
      <div>Olá</div>
    </Slide>
  );
}
```

Enable color on dark

Following the Material Design guidelines, the `color` prop has no effect on the appearance of the app bar in dark mode. You can override this behavior by setting the `enableColorOnDark` prop to `true`.

{{"demo": "EnableColorOnDarkAppBar.js", "bg": true}}