

## Sobre o lab

Este pacote hospeda os componentes da incubadora que ainda não estão prontos para mover para o core.

A principal diferença entre o lab e o core (núcleo) é como os componentes são versionados. Ter um pacote separado para o lab nos permite liberar alterações críticas quando necessário, enquanto o pacote core segue uma política de liberação mais controlada.

À medida que os desenvolvedores usam, testam os componentes e relatam problemas, os mantenedores aprendem mais sobre as deficiências dos componentes: recursos ausentes, problemas de acessibilidade, bugs, design de API, etc. Quanto mais antigo e mais usado for um componente, menor é a probabilidade de novos problemas serem encontrados e, conseqüentemente, necessidades de alterações críticas serem feitas. The older and more used a component is, the less likely it is that new issues will be found and subsequently need to introduce breaking changes.

Para que um componente esteja pronto para ir para o core, são considerados os seguintes critérios:

- Ele precisa ser **utilizado**. The Material-UI team uses Google Analytics in the documentation (among other metrics) to evaluate the usage of each component. Um componente do lab com baixo uso significa que ainda não está totalmente funcional, ou que há uma baixa demanda por ele.
- Ele precisa atender critérios de **qualidade de código**, semelhante aos componentes do core. O componente não precisa ser perfeito para fazer parte do core, mas ele deve ser confiável o suficiente para que os desenvolvedores possam depender dele.
  - Cada componente precisa de **definições de tipo**. Atualmente, não é necessário que um componente do lab seja tipado, mas ele precisará ser tipado para passar para o core.
  - Requer boa **cobertura de teste**. Atualmente, alguns dos componentes do lab não têm testes abrangentes.
- O componente pode ser usado como **alavancagem** para incentivar os usuários a atualizar para a versão mais recente? Quanto menos fragmentada a comunidade, melhor.
- Ele precisa ter uma baixa probabilidade de **alterações críticas** no curto/médio prazo. Por exemplo, se o componente precisar de um novo recurso que provavelmente precisará de alterações críticas, pode ser preferível adiar sua promoção para o core.

## Instalação

Instale o pacote no diretório do seu projeto com:

```
// usando npm
npm install @mui/lab
```

```
// usando yarn
yarn add @mui/lab
```

O lab tem dependências com os componentes do core. Se você ainda não está utilizando o MUI em seu projeto, poderá instalá-lo com:

```
// usando npm
npm install @mui/material
```

```
// usando yarn
yarn add @mui/material
```

## TypeScript

Para se beneficiar de CSS overrides e customização de propriedades padrão com o tema, usuários de TypeScript precisam importar os seguintes tipos. Internamente, ele usa ampliação de módulos para estender a estrutura padrão do tema com os componentes de extensão disponíveis no lab.

```
// Quando utilizando TypeScript 4.x e acima
import type {} from '@mui/lab/themeAugmentation';
// Quando utilizando TypeScript 3.x e abaixo
import '@mui/lab/themeAugmentation';
```

```
const theme = createTheme({
  components: {
    MuiTimeline: {
      styleOverrides: {
        root: {
          backgroundColor: 'red',
        },
      },
    },
  },
});
```