

SLIDEV Docs - Português-BR



Table of contents

• Br - README	4
• Br - TRANSLATIONS	5
• Builtin - Components	8
• Builtin - Layouts	9
• Custom - Config katex	13
• Custom - Config mermaid	14
• Custom - Config monaco	15
• Custom - Config shortcuts	17
• Custom - Config vite	18
• Custom - Config vue	19
• Custom - Config windicss	20
• Custom - Directory structure	21
• Custom - Fonts	25
• Custom - Global layers	28
• Custom - Highlighters	30
• Custom - Index	32
• Custom - Vue context	34
• Guide - Animations	36
• Guide - Drawing	40
• Guide - Editors	42
• Guide - Exporting	45
• Guide - Faq	46
• Guide - Hosting	49
• Guide - Index	52
• Guide - Install	55
• Guide - Navigation	57
• Guide - Presenter mode	59
• Guide - Recording	60
• Guide - Syntax	61
• Guide - Why	70
• Resources - Covers	73
• Resources - Learning	74
• Br - Showcases	75
• Themes - Gallery	76

- Themes - Use _____ 77
- Themes - Write a /theme _____ 79

sli.dev

Documentação do [Slidev](#)

Traduções

	Repositório	Site	Mantenedores
English	docs	sli.dev	@antfu
简体中文	docs-cn	cn.sli.dev	@QC-L @Ivocin
Français	docs-fr	fr.sli.dev	@ArthurDanjou
Español	docs-es	es.sli.dev	@owlnai
Русский	docs-ru	ru.sli.dev	@xesjkeee
Việt Nam	docs-vn	vn.sli.dev	@bongudth
Deutsch	docs-de	de.sli.dev	@fabiankachlock
Português (BR)	docs-br	br.sli.dev	@luisfelipesdn12
Ελληνικά	docs-el	el.sli.dev	@GeopJr

Rode o servidor localmente

```
npm i -g pnpm
pnpm i
pnpm run dev
```

E depois vá para <http://localhost:3000>

Ou instale a [extensão Vite para VS Code](#) pra editar lado a lado.

Ajude a Traduzir

Veja [TRANSLATIONS.md](#)

[Go to TOC](#)

Help on Translating

First of all, thank you for being interested in contributing to translations!

You can find the repositories for each existing translation in [README.md](#). To help improve them, simply sending a Pull Request to their repo.

If the language you want to contribute isn't on the list, join [our Discord server](#), and find the `#translations` channel to see if someone is already working on the language you want, consider joining them and translate together. If not, you can start a new translation project with the following steps.

In case it's already been translated but you're wondering how to maintain it, skip to the end. ## Some tips before you get started

- It is recommended that you use your IDE of choice (e.g VSCode) paired with a development server running, so you can see your translation changes in real-time.
- You can mark these checkmarks as the translation progresses or use your own workflow. The translations don't need to be made in any particular order.
- Translations don't need to be literal, but they should convey the same message. In case you're not sure how to translate something, you can either leave it as it is or use online tools like WordReference or Linguee to aid you.
- Most translations will simply consist in editing Markdown files. Certain areas are buried under Vue components, which will be listed below. You can also use your IDE to find the string to translate.

Getting started

- Fork the main docs repo: [slidevjs/docs](#)
- Translate README.md, you can take one of the already translated repositories as an example.
- Share your repo's link to the `#translations` channel telling people you are working on it and find collaborators.

Translating Markdown files

- `showcases.md` - A gallery showcase of Slidev presentations.
- `index.md` - Mainpage content, note that some of it is buried under Vue components listed further below.

.vitepress/

- `config.js` - Sitemap
- `/theme/components/HomeHero.vue` - The hero from the mainpage
- `/theme/components/WorkingInProgress.vue` - WIP notice shown in mainpage
- `/theme/components/demo/Demo.vue` - Animated demo shown in mainpage
- `/theme/components/Environment.vue` - Describes the environment of a setting.

builtin/

- `components.md` - Use [Vue components](#) inside Slidev
- `layouts.md` - Use Vue layouts inside Slidev

custom/

- `config-katex.md` - Configuring Katex
- `config-mermaid.md` - Configuring Mermaid
- `config-monaco.md` - Configuring Monaco
- `config-shortcuts.md` - Configuring Shortcuts
- `config-vite.md` - Configuring Vite
- `config-vue.md` - Configuring Vue
- `config-windicss.md` - Configuring Windicss
- `directory-structure.md` - Configuring the directory structure
- `fonts.md` - Configuring fonts
- `global-layers.md` - Configuring the global layers
- `highlighters.md` - Configuring code highlighters
- `index.md` - Customizations index page
- `vue-context.md` - The Vue global context

guide/

- `animations.md` - Animations and transitions
- `editors.md` - Editor integrations
- `exporting.md` - Exporting your slides
- `faq.md` - Frequent Answered Questions
- `index.md` - Getting started with Slidev
- `install.md` - Installing Slidev
- `navigation.md` - Navigation across slides
- `presenter-mode.md` - Toggling presenter mode
- `recording.md` - Recording your presentation
- `syntax.md` - Markdown syntax
- `why.md` - *Why Slidev?*

resources/

- `covers.md` - Curated covers for Slidev

themes/

- `gallery.md` - Theme gallery
- `use.md` - How to use Slidev themes
- `write-a-theme.md` - Write your own theme

Publishing your translations

- When you finish the translation (at least 90%), `@antfu` in the Discord and we will invite you to the org and make the translation official.
- Once the transferring is done, we will set up the subdomain, auto-deployment, and a daily sync-up bot to keep the translation up-to-date with the latest English docs.
- The site is live, and we will send a shout-out tweet on [our Twitter account](#).

Maintaining the translations up-to-date

- `docschina-bot` will periodically submit merge requests from the `slidev/docs` repository. Switch to the branch created in the pull request, make any changes necessary and merge it. [example](#).
- Sometimes it will occur that a merge request is made and you haven't merged the previous one. The latest PR always checks your main branch against the English one; so you can just close the previous PR(s), move your work to the latest one and merge it.

[Working-in-progress translation list](#)

Thanks again!

[Go to TOC](#)

Componentes

Componentes internos

A documentação dessa seção continua em progresso. Até lá, você pode olhar o [código-fonte](#) diretamente.

Componentes customizados

Crie um diretório `components/` no diretório raiz do projeto, e simplesmente insira seu componente customizado do Vue lá dentro, depois disso você poderá usá-lo com o mesmo nome no seu arquivo markdown!

Leia mais na seção [Customização](#)

Componentes providos por temas

Temas também podem fornecer componentes. É recomendado ler a documentação do tema para ver o que eles provém.

Saiba mais na seção de [estrutura de diretório](#).

[Go to TOC](#)

Layouts

Layouts Internos

Já que temas podem sobrepor o comportamento dos layouts, o melhor jeito de saber exatamente o uso, parâmetros e exemplos é consultando suas respectivas documentações.

center

Exibe o conteúdo no meio da tela.

cover

Usado para exibir a página de capa para a apresentação, pode conter o título, contextualização, etc.

default

O layout mais básico, para exibir qualquer conteúdo.

end

A página final da apresentação.

fact

Para mostrar algum fato ou dado com bastante destaque na tela.

full

Usa todo o espaço da tela para exibir o conteúdo.

image-left

Mostra uma imagem no lado esquerdo da tela, o conteúdo será posicionado à direita.

Uso

```
---  
layout: image-left  
  
# a fonte da imagem  
image: ./caminho/para/a/imagem  
  
# uma nome de classe customizado para o conteúdo  
class: my-cool-content-on-the-right  
---
```

image-right

Mostra uma imagem no lado direito da tela, o conteúdo será posicionado à esquerda.

Uso

```
---
layout: image-right
# a fonte da imagem
image: ./caminho/para/a/imagem
# uma nome de classe customizado para o conteúdo
class: my-cool-content-on-the-left
---
```

image

Mostra uma imagem como o conteúdo principal da página.

Uso

```
---
layout: image
# a fonte da imagem
image: ./caminho/para/a/imagem
---
```

iframe-left

Shows a web page on the left side of the screen, the content will be placed on the right side.

Usage

```
---
layout: iframe-left
# the web page source
url: https://github.com/slidesjs/slides
# a custom class name to the content
class: my-cool-content-on-the-right
---
```

iframe-right

Shows a web page on the right side of the screen, the content will be placed on the left side.

Usage

```
---
layout: iframe-right
# the web page source
url: https://github.com/slidesjs/slides
```

```
--  
# a custom class name to the content  
class: my-cool-content-on-the-left  
---
```

iframe

Shows a web page as the main content of the page.

Usage

```
--  
layout: iframe  
  
# the web page source  
url: https://github.com/slidesjs/slides  
---
```

intro

Para introduzir a apresentação, geralmente com o título, uma descrição curta, o autor, etc.

none

Um layout sem nenhum estilo existente.

quote

Para exibir uma citação com destaque.

section

Usado para marcar o início de uma nova seção da apresentação.

statement

Fazer uma afirmação/declaração como o principal conteúdo da página.

two-cols

Separa o conteúdo da página em duas colunas.

Uso

```
--  
layout: two-cols  
--  
  
# Esquerda  
  
Isto está à esquerda  
::right::
```

Direita

Isto está à direita

Layouts Customizados

Crie um diretório `layouts/` no diretório raiz do projeto, e simplesmente insira os componentes do seu layout customizado do Vue lá dentro.

Leia mais na seção [Customização](#)

Layouts provados por temas

Temas podem fornecer layouts ou até mesmo sobrepor layouts existentes. É recomendado ler a documentação do tema para ver o que eles provém.

[Go to TOC](#)

Configurar KaTeX

Crie o arquivo `./setup/katex.ts` com o seguinte conteúdo:

```
import { defineKatexSetup } from '@slidev/types'

export default defineKatexSetup(() => {
  return {
    /* ... */
  }
})
```

Com essa configuração, você pode prover as configurações customizadas para o [KaTeX Options](#). Consulte as definições de tipo e suas documentações para obter mais detalhes.

Configurar Mermaid

Crie o arquivo `./setup/mermaid.ts` com o seguinte conteúdo:

```
import { defineMermaidSetup } from '@slidev/types'

export default defineMermaidSetup(() => {
  return {
    theme: 'forest',
  }
})
```

Com essa configuração, você pode prover as configurações customizadas para o [Mermaid](#). Consulte as definições de tipo e suas documentações para obter mais detalhes.

Configurar Monaco

Crie o arquivo `./setup/monaco.ts` com o seguinte conteúdo:

```
import { defineMonacoSetup } from '@slidev/types'

export default defineMonacoSetup(async (monaco) => {
  // use `monaco` to configure
})
```

Saiba mais sobre [como configurar o Monaco](#).

Uso

Para usar o Monaco nos seus slides, simplesmente acrescente `{monaco}` aos seus blocos de código:

```
//``js
const count = ref(1)
const plusOne = computed(() => count.value + 1)

console.log(plusOne.value) // 2

plusOne.value++ // error
//``
```

Para

```
//``js {monaco}
const count = ref(1)
const plusOne = computed(() => count.value + 1)

console.log(plusOne.value) // 2

plusOne.value++ // error
//``
```

Exportando

Por padrão, Monaco só vai funcionar no modo `dev`. Se quiser tê-lo na aplicação SPA exportada, configure isso no seu frontmatter:

```
---
monaco: true # padrão "dev"
---
```

Instalação Automática dos Tipos

Quando é usado TypeScript com o Monaco, os tipos para as dependências serão instalados automaticamente no lado do cliente.

```
//``ts {monaco}
import { ref } from 'vue'
import { useMouse } from '@vueuse/core'

const counter = ref(0)
//``
```

Neste exemplo acima, verifique que tanto o `vue` quanto o `@vueuse/core` estão instalados localmente assim como as dependências / dependências dev, o Slidev vai lidar com o resto para obter os tipos para o editor automaticamente!

Configurar Temas

O tema é controlado pelo Slidev baseado no modo claro/escuro. Se você quiser customizar-lo, é só passar o id do tema na função de configuração:

```
// ./setup/monaco.ts
import { defineMonacoSetup } from '@slidev/types'

export default defineMonacoSetup(() => {
  return {
    theme: {
      dark: 'vs-dark',
      light: 'vs',
    },
  }
})
```

Se você quiser carregar temas personalizados:

```
import { defineMonacoSetup } from '@slidev/types'

// change to your themes
import dark from 'theme-vitesse/themes/vitesse-dark.json'
import light from 'theme-vitesse/themes/vitesse-light.json'

export default defineMonacoSetup((monaco) => {
  monaco.editor.defineTheme('vitesse-light', light as any)
  monaco.editor.defineTheme('vitesse-dark', dark as any)

  return {
    theme: {
      light: 'vitesse-light',
      dark: 'vitesse-dark',
    },
  }
})
```

Se você estiver criando um tema para o Slidev, use o `import()` dinâmico na função de configuração para obter resultados melhores.

[Go to TOC](#)

Configurar Atalhos

Disponível a partir da v0.20

Crie o arquivo `./setup/shortcuts.ts` com o seguinte conteúdo:

```
import { defineShortcutsSetup, NavOperations } from '@slidev/types'

export default defineShortcutsSetup((nav: NavOperations) => {
  return [
    {
      key: 'enter',
      fn: () => nav.next(),
      autoRepeat: true,
    },
    {
      key: 'backspace',
      fn: () => nav.prev(),
      autoRepeat: true,
    },
  ]
})
```

Com a configuração, você pode fornecer as customizações para os atalhos mencionados em [Navegação](#). A configuração acima vincula a próxima animação ou slide ao enter e a animação ou slide anterior ao backspace.

A função de configuração recebe um objeto com alguns métodos de navegação, e retorna um array contendo alguns atalhos de configuração. Confira as definições de tipos para mais detalhes.

Confira [useMagicKeys](#) | [VueUse](#) para mais detalhes sobre o evento de uma tecla pressionada.

Configurar Vite

O Sliderv usa o [Vite](#) por baixo dos panos. Isso significa que você pode aproveitar o excelente sistema de plug-ins do Vite para personalizar ainda mais seus slides.

O arquivo `vite.config.ts` será seguido se você tiver um.

O Sliderv possui os seguintes plug-ins pré-configurados:

- [@vitejs/plugin-vue](#)
- [unplugin-vue-components](#)
- [unplugin-icons](#)
- [vite-plugin-md](#)
- [vite-plugin-windicss](#)
- [vite-plugin-remote-assets](#)

Saiba mais sobre as [pré-configurações aqui](#).

Configurar Plug-ins Internos

Disponível a partir da v0.21

Para configurar os plug-ins listados acima, crie um arquivo `vite.config.ts` com o seguinte conteúdo. Se atente ao fato de que o Sliderv tem algumas opções pré-configuradas para estes plug-ins, e esse uso vai sobrepor algumas delas, o que potencialmente pode quebrar a aplicação. Por favor, trate isto como uma **funcionalidade avançada**, certifique-se de que você sabe o que está fazendo antes de continuar.

```
import { defineConfig } from 'vite'

export default defineConfig({
  slidev: {
    vue: {
      /* opções do vue */
    },
    markdown: {
      /* opções do markdown-it */
      markdownItSetup(md) {
        /* plug-ins customizados do markdown-it */
        md.use(/* ... */)
      },
    },
    /* opções para outros plug-ins */
  },
})
```

Veja as [declarações de tipo](#) para mais opções.

[Go to TOC](#)

Configurar Vue

O Slidev usa o [Vue 3](#) para renderizar a aplicação no lado do cliente. Você pode estender a aplicação para adicionar plug-ins e configurações customizadas.

Crie o arquivo `./setup/main.ts` com o seguinte conteúdo:

```
import { defineAppSetup } from '@slidev/types'

export default defineAppSetup(({ app, router }) => {
  // Vue App
  app.use(YourPlugin)
})
```

Isto pode ser usado também como a entrada principal da sua aplicação do Slidev para fazer algumas inicializações antes da aplicação rodar.

Saiba mais: [Vue Application API](#).

Configurar Windi CSS

O Markdown suporta naturalmente tags HTML embutidas. Portanto, você pode estilizar seu conteúdo como preferir. Para oferecer certa comodidade, nós temos o [Windi CSS](#) embutido, para que você possa estilizar o conteúdo diretamente usando classes.

Por exemplo:

```
<div class="grid pt-4 gap-4 grids-cols-[100px,1fr]>
  ### Nome
  - Item 1
  - Item 2
</div>
```

O [Attributify Mode](#) no [Windi CSS v3.0](#) é habilitado por padrão.

Configurações

Para configurar o Windi CSS, crie o arquivo `setup/windicss.ts` com o seguinte conteúdo para estender as configurações existentes:

```
// setup/windicss.ts

import { defineWindiSetup } from '@slidev/types'

// extendendo a configuração existente do windicss
export default defineWindiSetup(() => ({
  shortcuts: {
    // plano de fundo padrão customizado
    'bg-main': 'bg-white text-[#181818] dark:(bg-[#121212] text-[#ddd])',
  },
  theme: {
    extend: {
      // fontes podem ser substituídas aqui, lembre-se de atualizar os links de
      // fontes da web no `index.html`
      fontFamily: {
        sans: 'ui-sans-serif,system-ui,-apple-system,BlinkMacSystemFont,"Segoe UI",Roboto,"Helvetica Neue",Arial,"Noto Sans",sans-serif,"Apple Color Emoji","Segoe UI Emoji","Segoe UI Symbol","Noto Color Emoji"',
        mono: '"Fira Code", monospace',
      },
    },
  },
}))
```

Saiba mais sobre as [configurações do Windi CSS](#).

[Go to TOC](#)

Estrutura de Diretório

O Slidev emprega algumas convenções de estrutura de diretório para minimizar a superfície de configuração e tornar as extensões das funcionalidades flexíveis e intuitivas.

A estrutura básica é a seguinte:

```
seu-slidev/
  ├── components/      # componentes customizados
  ├── layouts/         # layouts customizados
  ├── public/          # recursos estáticos
  ├── setup/           # configurações / hooks customizados
  ├── styles/          # estilos customizados
  ├── index.html       # injeções ao index.html
  ├── slides.md        # a entrada principal dos slides
  └── vite.config.ts   # extensões às configurações do vite
```

Todos eles são opcionais.

Componentes

Convenções: `./components/*.{vue,js,ts,jsx,tsx,md}`

Os componentes dentro deste diretório podem ser usados diretamente no Markdown dos slides como o mesmo nome do arquivo.

Por exemplo:

```
seu-slidev/
  ...
  └── components/
      └── MeuComponente.vue
      └── HelloWorld.ts
```

```
<!-- slides.md -->

# Meu slide

<MeuComponente :count="4"/>

<!-- os dois nomes funcionam -->

<hello-world foo="bar">
  Slot
</hello-world>
```

Esta funcionalidade é fornecida pelo `vite-plugin-components`, saiba mais por lá.

O Slidev também disponibiliza alguns [componentes embutidos](#) pra você usar.

Layouts

Convenções: `./layouts/*.{vue,js,ts,jsx,tsx}`

```
su-slidev/
  ...
  └── layouts/
    ├── cover.vue
    └── meu-tema-supimpa.vue
```

Você pode usar qualquer nome de arquivo para o seu layout. Depois é só referenciar o layout no seu cabeçalho YAML usando o nome do arquivo.

```
---  
layout: meu-tema-supimpa  
---
```

Se o layout que você fez tiver o mesmo nome de um interno ou de algum tema, o seu layout customizado terá prioridade sobre os demais. A ordem de prioridade é `local > tema > interno`.

No componente de layout, use `<slot/>` para o conteúdo do slide. Por exemplo:

```
<!-- default.vue -->
<template>
  <div class="slidev-layout default">
    <slot />
  </div>
</template>
```

Público

Convenções: `./public/*`

Recursos neste diretórios serão servidos na rota raiz `/` durante o desenvolvimento, e copiados para a raiz do diretório `dist` do jeito que estiverem. Leia mais sobre [o diretório `public` do Vite's](#).

Estilo

Convenções: `./style.css | ./styles/index.{css,js,ts}`

Arquivos que seguem esta convenção serão injetados na raiz da aplicação. Se você precisar importar múltiplas entradas de CSS, você pode criar a seguinte estrutura e administrar a ordem de importação por conta própria.

```
seu-slidev/
  ...
  └── styles/
    ├── index.ts
    ├── base.css
    ├── code.css
    └── layouts.css
```

```
// styles/index.ts

import './base.css'
import './code.css'
import './layouts.css'
```

Estilos serão processados pelo [Windi CSS](#) e o [PostCSS](#), portanto você pode usar aninhamento de css e as [at-directives](#) sem nenhuma configuração adicional. Por exemplo:

```
.slidev-layout {
  @apply px-14 py-10 text-[1.1rem];

  h1, h2, h3, h4, p, div {
    @apply select-none;
  }

  pre, code {
    @apply select-text;
  }

  a {
    color: theme('colors.primary');
  }
}
```

[Aprenda mais sobre a sintaxe.](#)

index.html

Convenções: `index.html`

O `index.html` provê a habilidade de injetar meta tags e/ou script ao arquivo `index.html` principal.

Por exemplo, no caso do seguinte `index.html` customizado:

```
<!-- ./index.html -->
<head>
  <link rel="preconnect" href="https://fonts.gstatic.com">
  <link href="https://fonts.googleapis.com/css2?family=Fira+Code:wght@400;600&family=Nunito+Sans:wght@200;400;600&display=swap" rel="stylesheet">
</head>

<body>
  <script src=".//seus-scripts"></script>
</body>
```

O `index.html` final hospedado será:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="icon" type="image/png"
  href="https://cdn.jsdelivr.net/gh/slidevjs/slidev/assets/favicon.png">
  <!-- injected head -->
```

```
<link rel="preconnect" href="https://fonts.gstatic.com">
<link href="https://fonts.googleapis.com/css2?
family=Fira+Code:wght@400;600&family=Nunito+Sans:wght@200;400;600&display=swap"
rel="stylesheet">
</head>
<body>
  <div id="app"></div>
  <script type="module" src="__ENTRY__"></script>
  <!-- injected body -->
  <script src="./seu-scripts"></script>
</body>
</html>
```

Camadas Globais

Convenções: [global-top.vue](#) | [global-bottom.vue](#)

Saiba mais: [Camadas Globais](#)

[Go to TOC](#)

Fontes

Disponível a partir da v0.20

Enquanto você pode usar HTML e CSS para customizar as fontes e estilos nos seus slides como preferir, o Slidev também provê um jeito conveniente de usá-los sem esforços.

No seu frontmatter, configure assim:

```
---  
  fonts:  
    # basicamente o texto  
    sans: 'Robot'  
    # use com a classe `font-serif` do windicss  
    serif: 'Robot Slab'  
    # para blocos de código, códigos inline, etc.  
    mono: 'Fira Code'  
---
```

E é só isso.

As fontes serão **importadas automaticamente do Google Fonts**. Isso significa que você está livre para usar qualquer fonte disponível no Google Fonts diretamente.

Fontes Locais

Por padrão, o Slidev assume que todas as fontes especificadas por meio da configuração `fonts` vêm do Google Fonts. Se você quiser usar fontes locais, especifique o `fonts.local` para interromper a importação automática.

```
---  
  fonts:  
    # assim como no font-family do css, você pode usar `,` para separar múltiplas  
    # como substitutas  
    sans: 'Helvetica Neue, Robot'  
    # marca 'Helvetica Neue' como uma fonte local  
    local: 'Helvetica Neue'  
---
```

Espessura & Itálico

Por padrão, o Slidev importa três espessuras `200, 400, 600` para cada fonte. Você pode configurá-las assim:

```
---  
  fonts:  
    sans: 'Robot'  
    # padrão  
    weights: '200,400,600'
```

```
# importa fontes itálicas, o padrão é `false`  
italic: false
```

Esta configuração se aplica a todas as fontes da web. Para um controle mais refinado da espessura de cada fonte, você vai precisar importá-las manualmente com [HTML](#) e CSS.

Fontes Substitutas

Na maioria das vezes, você só vai precisar especificar a "fonte especial" e o Slidev vai incluir as fontes substitutas para você, por exemplo:

```
---  
fonts:  
  sans: 'Robot'  
  serif: 'Robot Slab'  
  mono: 'Fira Code'  
---
```

vai resultar em

```
.font-sans {  
  font-family: "Robot", ui-sans-serif, system-ui, -apple-  
  system, BlinkMacSystemFont, "Segoe UI", Roboto, "Helvetica Neue", Arial, "Noto  
  Sans", sans-serif, "Apple Color Emoji", "Segoe UI Emoji", "Segoe UI Symbol", "Noto  
  Color Emoji";  
}  
.font-serif {  
  font-family: "Robot Slab", ui-serif, Georgia, Cambria, "Times New  
  Roman", Times, serif;  
}  
.font-mono {  
  font-family: "Fira Code", ui-monospace, SFMono-  
  Regular, Menlo, Monaco, Consolas, "Liberation Mono", "Courier New", monospace;  
}
```

Se você quiser desabilitar as fontes substitutas, configure assim:

```
---  
fonts:  
  mono: 'Fira Code, monospace'  
  fallback: false  
---
```

Provedores

- Opções: `google` | `none`
- Padrão: `google`

Atualmente, apenas o Google Fonts é suportado, nós planejamos adicionar mais provedores futuramente. Especificar para `none` desabilitará a funcionalidade de importação automática por completo e fará com que todas as fontes sejam tratadas como fontes locais.

```
---  
  fonts:  
    provide: 'none'  
---
```

Camadas Globais

Disponível a partir da v0.17

Camadas globais te permitem ter componentes customizados que **permanecem** pelos slides. Isto pode ser útil para ter rodapés, animações entre os slides, efeitos globais, etc.

O Slidev disponibiliza duas camadas para está funcionalidade, crie o arquivo `global-top.vue` ou `global-bottom.vue` na raiz do projeto e elas serão inseridas automaticamente.

Relacionamento das Camadas:

- Cabeçalho Global (`global-top.vue`)
- Slides
- Rodapé Global (`global-bottom.vue`)

Exemplo

```
<!-- global-top.vue -->
<template>
  <footer class="absolute bottom-0 left-0 right-0 p-2">Seu Nome</footer>
</template>
```

O texto `Seu Nome` vai aparecer em todos os slides.

Para habilitar isso em função de alguma condição, você pode utilizar com o [Vue Global Context](#).

```
<!-- oculta o rodapé na Página 4 -->
<template>
  <footer
    v-if="$slidev.nav.currentPage !== 4"
    class="absolute bottom-0 left-0 right-0 p-2">
    >
      Seu Nome
    </footer>
  </template>
```

```
<!-- oculta o rodapé do layout "cover" -->
<template>
  <footer
    v-if="$slidev.nav.currentLayout !== 'cover'"
    class="absolute bottom-0 left-0 right-0 p-2">
    >
      Seu Nome
    </footer>
  </template>
```

```
<!-- um exemplo de rodapé para páginas -->
<template>
  <footer
```

```
v-if="$slidev.nav.currentLayout !== 'cover'"  
  class="absolute bottom-0 left-0 right-0 p-2"  
>    {{ $slidev.nav.currentPage }} / {{ $slidev.nav.total }}  
  </footer>  
</template>
```

Destacadores

O Slidev vem com dois destacadores de sintaxe pra você escolher:

- [Prism](#)
- [Shiki](#)

Prism é um dos destacadores de sintaxe mais populares. O destaque é feito adicionando classes de token ao código e é colorido utilizando CSS. Você pode pesquisar nos seus [temas oficiais](#), ou criar/customizar um você mesmo de forma bem fácil usando o `prism-theme-vars`.

Shiki, por outro lado, é um destacador de sintaxe TextMate baseado em gramática. Ele gera tokens coloridos, então não é necessário CSS adicional. Já que ele tem um ótimo suporte gramatical, as cores geradas são bem precisas, como as que você vê no VS Code. Shiki também vem com [vários temas predefinidos](#). O ponto negativo do Shiki é que ele também precisa de temas TextMate (compatíveis com temas do VS Code) para fazer o destaque, o que pode ser um pouco mais complicado de customizar.

Os temas do Slidev geralmente suportam tanto Prism quanto Shiki, mas dependendo do temas que estiver usando, pode ser que só suporte um deles.

Quando você tiver escolha, a troca é basicamente:

- **Prism** para uma customização mais fácil
- **Shiki** para um destaque mais preciso

Por padrão, o Slidev usa Prism. Você pode mudar isso modificando seu frontmatter:

```
---  
highlighter: shiki  
---
```

Configurar Prism

Para configurar seu Prism, você pode simplesmente importar o tema css or usar o `prism-theme-vars` para configurar temas tanto para o modo claro quanto para o escuro. Consulte suas documentações para mais detalhes.

Configurar Shiki

Crie o arquivo `./setup/shiki.ts` com o seguinte conteúdo:

```
/* ./setup/shiki.ts */  
import { defineShikiSetup } from '@slidev/types'  
  
export default defineShikiSetup(() => {  
    return {  
        theme: {  
            dark: 'min-dark',  
            light: 'min-light',  
        },  
    };  
});
```

```
  },
})
```

Consulte a [documentação do Shiki](#) para ver o nome dos temas disponíveis.

Ou se você quiser usar seu próprio tema:

```
/* ./setup/shiki.ts */

import { defineShikiSetup } from '@slidev/types'

export default defineShikiSetup(async({ loadTheme }) => {
  return {
    theme: {
      dark: await loadTheme('caminho/para/tema.json'),
      light: await loadTheme('caminho/para/tema.json'),
    },
  }
})
```

Customizações

O Slidev é totalmente customizável, desde estilização até configurações de ferramentas. Ele te permite configurar as ferramentas por trás ([Vite](#), [Windi CSS](#), [Monaco](#), etc.)

Configurações do Frontmatter

Você pode configurar o Slidev no frontmatter do seu primeiro slide, abaixo é mostrado os valores padrão para cada opção.

```
---  
# id fo tema ou nome do módulo  
theme: 'default'  
# título do seu slide, será automaticamente inferido do primeiro header se não for espeecificado  
title: ''  
# titleTemplate para a página web, `%-s` será substituído pelo título da página  
titleTemplate: '%s - Slidev'  
  
# habilita o download em PDF na SPA compilada, também pode ser uma url customizada  
download: true  
# destacador de sintaxe, pode ser 'prism' ou 'shiki'  
highlighter: 'prism'  
# mostra o número das linhas nos blocos de código  
lineNumbers: false  
# habilita o editor monaco, por padrão é só no dev  
monaco: 'dev'  
  
# força um esquema de cor aos slides, pode ser 'auto', 'light' ou 'dark'  
colorSchema: 'auto'  
# modo de roteamento para o vue-router, pode ser 'history' ou 'hash'  
routerMode: 'history'  
# proporção da tela para os slides  
aspectRatio: '16/9'  
# largura real da tela, unidade em px  
canvasWidth: 980  
  
# fontes serão automaticamente importadas do Google fonts  
# Saiba mais: https://sli.dev/custom/fonts  
fonts:  
  sans: 'Roboto'  
  serif: 'Roboto Slab'  
  mono: 'Fira Code'  
  
# frontmatter padrão aplicado para todos os slides  
defaults:  
  layout: 'default'  
  # ...  
  
# informações para seus slides, pode ser uma string em markdown  
info: |  
  ## Slidev  
  Minha primeira apresentação com o [Slidev](http://sli.dev/)!  
---
```

Vejas as [definições de tipos](#) para mais opções.

Estrutura de Diretório

O Slidev emprega algumas convenções de estrutura de diretório para minimizar a superfície de configuração e tornar as extensões das funcionalidades flexíveis e intuitivas.

Consulte a seção [Estrutura de Diretório](#).

Configurar Ferramentas

- [Destacadores](#)
- [Configurar Vue](#)
- [Configurar Vite](#)
- [Configurar Windi CSS](#)
- [Configurar Monaco](#)
- [Configurar KaTeX](#)
- [Configurar Mermaid](#)

Contexto Global do Vue

O Slidev injeta um contexto global do Vue `$slidev` para condições avançadas ou controles de navegação.

Uso

Você pode acessá-lo de qualquer lugar no seu markdown e template Vue, com a sintaxe "Mustache".

```
<!-- slides.md -->
# Página 1
A página atual é: {{ $slidev.nav.currentPage }}
```

```
<!-- Foo.vue -->
<template>
  <div>Título: {{ $slidev.configs.title }}</div>
  <button @click="$slidev.nav.next">Próxima Página</button>
</template>
```

Propriedades

`$slidev.nav`

Um objeto reativo contendo as propriedades e controles da navegação dos slides. Por exemplo:

```
$slidev.nav.next() // vai para o próximo passo
$slidev.nav.nextSlide() // vai para o próximo slide (pula v-clicks)
$slidev.nav.go(10) // vai para o slide #10
```

```
$slidev.nav.currentPage // número do slide atual
$slidev.nav.currentLayout // id do layout atual
$slidev.nav.clicks // contagem de clicks atual
```

Para mais propriedades disponíveis, consulte as exportações do arquivo `nav.ts`.

`$slidev.configs`

Um objeto reativo que contém as configurações do primeiro frontmatter do seu `slides.md`. Por exemplo:

```
---
title: Meu Primeiro Slidev!
---
```

```
{{ $slidev.configs.title }} // 'Meu Primeiro Slidev!'
```

\$slidev.themeConfigs

Um objeto reativo contendo as configurações de tema.

```
---  
title: Meu Primeiro Slidev!  
themeConfig:  
  primary: #213435  
---
```

```
{} $slidev.themeConfigs.primary } // '#213435'
```

Animações

Animações no clique

v-click

Para aplicar "animações no clique" para os elementos, você pode usar a diretriz `v-click` ou os componentes `<v-click>`.

```
# Olá
<!-- Uso do componente: isto ficará invisível até você pressionar "próximo" -->
<v-click>

Olá Mundo
</v-click>

<!-- Uso da diretriz: isto ficará invisível até você pressionar "próximo" pela
segunda vez -->
<div v-click class="text-xl p-2">

Eae!
</div>
```

v-after

O uso do `v-after` é similar ao do `v-click`, mas ele tornará o elemento visível quando o `v-click` anterior tiver sido disparado.

```
<div v-click>Olá</div>
<div v-after>Mundo</div>
```

Quando você clicar no botão "próximo", tanto `Olá` quanto `Mundo` aparecerão simultaneamente.

v-click-hide

Tem o mesmo comportamento do `v-click`, mas em vez de fazer o elemento aparecer, faz o elemento invisível após do clique.

```
<div v-click-hide>Olá</div>
```

v-clicks

`v-clicks` só é provido como um componente. É uma alternativa a aplicar a diretriz `v-click` em todos os elementos dentro dele. Ele é especialmente útil quando se está trabalhando com listas.

```
<v-clicks>
  - Item 1
```

```
- Item 2
- Item 3
- Item 4

</v-clicks>
```

Um item se tornará visível a cada vez que você clicar em "próximo".

Contagem de Cliques Customizadas

Por padrão, o Slidev calcula quantos passos são necessários antes de passar para o próximo slide. Você pode sobrepor esta configuração informando a propriedade `clicks` no frontmatter:

```
---  
# 10 cliques neste slide, antes de passar para o próximo  
clicks: 10  
---
```

Ordenação

Passando o índice o clique nas suas diretrizes, você pode customizar a ordem dos aparecimentos.

```
<div v-click>1</div>
<div v-click>2</div>
<div v-click>3</div>
```

```
<!-- a ordem é inversa -->
<div v-click="3">1</div>
<div v-click="2">2</div>
<div v-click="1">3</div>
```

```
---  
clicks: 3  
---  
  
<!-- visível após 3 cliques -->
<v-clicks at="3">
  <div>0i</div>
</v-clicks>
```

Transição de Elementos

Quando você aplica a diretriz `v-click` aos seus elementos, a eles serão anexados a classe `slidev-vclick-target`. Quando os elementos estão ocultos, a classe `slidev-vclick-hidden` também será anexada. Por exemplo:

```
<div class="slidev-vclick-target slidev-vclick-hidden">Texto</div>
```

Após o clique, isto se tornará

```
<div class="slidev-vclick-target">Texto</div>
```

Por padrão, uma transição suave de opacidade é aplicada à essas classes:

```
// o padrão

.slidev-vclick-target {
  transition: opacity 100ms ease;
}

.slidev-vclick-hidden {
  opacity: 0;
  pointer-events: none;
}
```

Você pode substituí-las no seu CSS customizado para alterar os efeitos de transição.

Por exemplo, você pode conseguir uma transição de aumentar a escala assim:

```
// styles.css

.slidev-vclick-target {
  transition: all 500ms ease;
}

.slidev-vclick-hidden {
  transform: scale(0);
}
```

Para especificar animações para um único slide ou layout definido:

```
.slidev-page-7,
.slidev-layout.my-custom-layout {
  .slidev-vclick-target {
    transition: all 500ms ease;
  }

  .slidev-vclick-hidden {
    transform: scale(0);
  }
}
```

Aprenda mais sobre a [customização de estilo](#)

Movimento

O Slidev tem o [@vueuse/motion](#) integrado. Você pode usar a diretriz `v-motion` em qualquer elemento para aplicar movimento a ele. Por exemplo:

```
<div
  v-motion
  :initial="{ x: -80 }"
  :enter="{ x: 0 }">
  Slidev
</div>
```

O texto `Slidev` se moverá de `-80px` até sua posição original na inicialização.

Nota: Slidev pré-carrega o próximo slide por performance, o que significa que as animações podem começar antes de você navegar até a página. Para fazer as animações funcionarem corretamente, você pode desabilitar o pré-carregamento no slide em particular.

```
---
```

```
preload: false
```

```
--
```

Ou manipular o ciclo de vida do elemento com o `v-if` para ter mais controle:

```
<div
```

```
  v-if="$slidev.nav.currentPage === 7"
```

```
  v-motion
```

```
  :initial="{ x: -80 }"
```

```
  :enter="{ x: 0 }">
```

```
    Slidev
```

```
</div>
```

Saiba mais: [Demonstração](#) | [@vueuse/motion](#) | [v-motion](#) | [Presets](#)

Transições de Página

O suporte integrado para transição entre slides AINDA NÃO é provido na versão atual. Estamos planejando adicionar o suporte na próxima versão principal. Até lá, você ainda pode usar estilos customizados e bibliotecas para fazer isso.

Desenhos & Anotações

Disponível a partir da v0.23

Nós temos o [drauu](#) embutido para desenhos e anotações que podem melhorar ainda mais suas apresentações.

Pra começar, clique no ícone na barra de navegação e comece a desenhar. Isto também está disponível no [Modo Apresentador](#). Desenhos e anotações que você criar serão **sincronizadas** automaticamente por todas as instâncias em tempo real.

Use com uma caneta Stylus

Quando usado com uma caneta stylus em um tablet (por exemplo, iPad com a Apple Pencil), o Slidev pode detectar com inteligência o tipo de entrada. Você pode desenhar diretamente nos seus slides com uma caneta sem precisar alternar para o modo de desenho, e tendo o controle de navegação com seus dedos ou mouse.

Manter Desenhos

A seguinte configuração do frontmatter te permite manter os desenhos como SVGs dentro do diretório `.slidev/drawings` e tê-los no seu PDF exportado site hospedado.

```
---  
  drawings:  
    persist: true  
---
```

Desabilitar Desenhos

Inteiramente:

```
---  
  drawings:  
    enabled: false  
---
```

Apenas no Desenvolvimento:

```
---  
  drawings:  
    enabled: dev  
---
```

Apenas no Modo de Apresentador:

```
---  
drawings:  
  presenterOnly: true  
---
```

Sincronização dos Desenhos

Por padrão, o Slidev sincroniza os seus desenhos por todas as instâncias. Se você estiver compartilhando seus slides com outras pessoas, você pode querer desabilitar a sincronização assim:

```
---  
drawings:  
  syncAll: false  
---
```

Com esta configuração, só os desenhos da instância do apresentador serão sincronizados com as outras.

Suporte em Editores

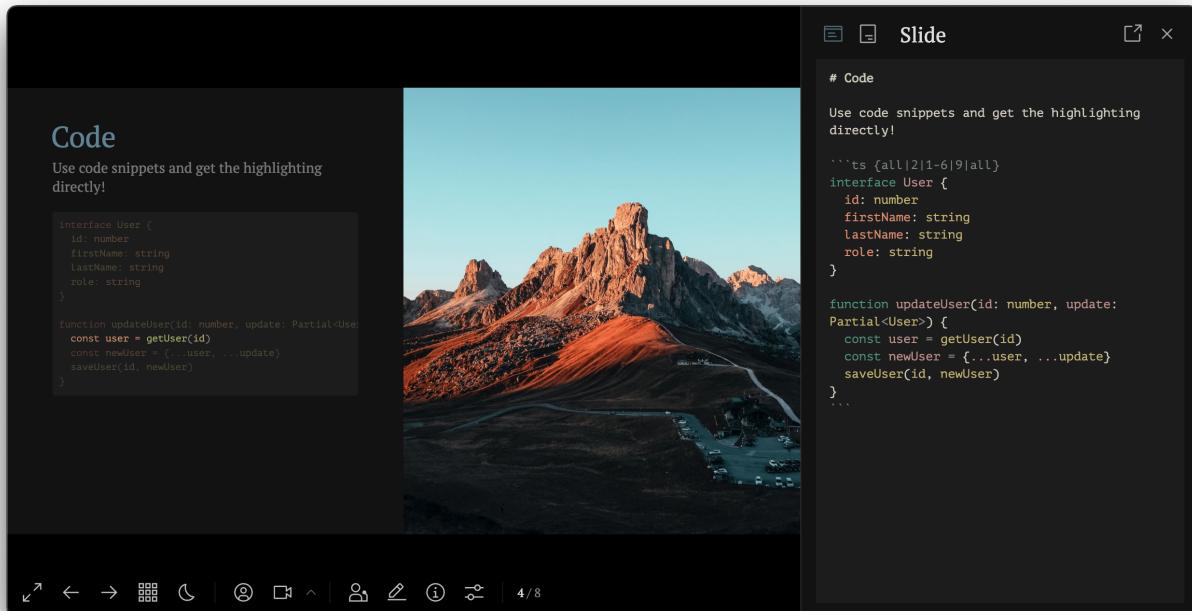
Já que o Slidev usa Markdown como fonte, você pode usar QUALQUER editor que ama escrever.

Se você quiser certo gerenciamento de alto-nível em seus slides, nós provemos as seguintes integrações em editores pra você!

Editor Integrado

O Slidev vem com um editor [CodeMirror](#) integrado que vai instantaneamente recarregar e salvar as alterações no seu arquivo.

Clique no botão para abri-lo.



Extensão do VS Code



Slidev for VS Code

VS Code Marketplace

v0.4.1

downloads 26k

A extensão do VS Code fornece algumas funcionalidades para te ajudar a organizar seus slides e ter uma breve visualização deles.

Funcionalidades

- Ver slides no painel lateral
- Botões de ir para o próximo / anterior
- Re-ordenação dos slides
- Separação de blocos no slide
- Conversão de Markdown para HTML

The screenshot shows a slide editor interface. On the left is a sidebar with a file tree titled "SLIDEV: SLIDES". The tree includes items like "Composable Vue", "Anthony Fu", "Sponsors", and several sections under "Composition API" such as "Ref", "Ref Auto Unwrapping", "Patterns & Tips", and "What's Composable Func...". On the right is a main editor window titled "slides.md — demo". The code in the editor is:

```
168 # Ref Auto Unwrapping <MarkerCore />
170 Get rid of `value` for most of the time.
<div class="grid grid-cols-2 gap-x-4">
<v-clicks :every='2'>
  - `watch` accepts ref as the watch target, and returns the
    unwrapped value in the callback
```
180 const counter = ref(0)

watch(counter, count => {
 console.log(count) // same as `counter.value`
})
```
  - Ref is auto unwrapped in the template
```
190 <template>
 <button @click="counter += 1">
```

---

[Go to TOC](#)

# Exportando

## PDF

A exportação para PDF depende do [Playwright](#) para renderizar. Você precisa instalar o `playwright-chromium` primeiro para usar esta funcionalidade. Se você estiver fazendo exportações num ambiente de CI, o [guia do playwright para CI](#) pode ser útil.

Instale o `playwright-chromium`

```
$ npm i -D playwright-chromium
```

Agora exporte seus slides para PDF usando o seguinte comando

```
$ slidev export
```

Após alguns segundos, seus slides estarão prontos em `./slides-exports.pdf`. [### Exportar etapas de cliques](#)

Disponível a partir da v0.21

Por padrão, o Slidev exporta uma página por slide com as animações de clique desabilitadas. Se você quiser exportar slides com múltiplas etapas entre múltiplas páginas, passe a opção `--with-clicks`.

```
$ slidev export --with-clicks
```

## PNGs

Ao passar a opção `--format png`, o Slidev vai exportar imagens em PNG para cada slide em vez de PDF.

```
$ slidev export --format png
```

## Single-Page Application (SPA)

Veja [Hospedagem Estática](#).

---

[Go to TOC](#)

# FAQ

## Grids

Já que o Slidev é baseado na Web, você pode aplicar qualquer layout de grid como quiser. [CSS Grids](#), [flexbox](#), ou até [Masonry](#), você tem total controle.

Como nós temos o [Windi CSS](#) integrado, aqui vai um jeito simples de você fazer isso:

```
<div class="grid grid-cols-2 gap-4">
<div>

A primeira coluna

</div>
<div>

A segunda coluna

</div>
</div>
```

Vá além, você pode customizar o tamanho de cada coluna assim:

```
<div class="grid grid-cols-[200px,1fr,10%] gap-4">
<div>

A primeira coluna (200px)

</div>
<div>

A segunda coluna (ajuste automático)

</div>
<div>

A terceira coluna (10% da largura do elemento pai)

</div>
</div>
```

Aprenda mais sobre [Windi CSS Grids](#).

## Posicionamento

Os slides são definidos em um tamanho fixo (padrão `980x552px`) e se ajusta para caber na tela do usuário. Você pode tranquilamente usar posições absolutas em seus slides e elas serão expandidas com a tela.

Por exemplo:

```
<div class="absolute left-30px bottom-30px">
Este é um rodapé alinhado no canto inferior esquerdo
</div>
```

Para alterar o tamanho do canvas, você pode passar a opção `canvasWidth` no seu primeiro frontmatter:

```

canvasWidth: 800

```

## Tamanho da Fonte

Se você sentir que o tamanho da fonte nos seus slides estão muito pequenas, você pode ajustá-lo de algumas formas:

### Substituir o Estilo Local

Você pode sobrepor estilos em cada slide com a tag `<style>`.

```
Página 1
<style>
h1 {
 font-size: 10em;
}
</style>

Página 2
Este aqui não será afetado.
```

Saiba mais: [Estilos Integrados](#)

### Substituir o Estilo Global

Você pode fornecer estilos globais customizados criando um arquivo `./style.css`, por exemplo:

```
/* style.css */
h1 {
 font-size: 10em !important;
}
```

Saiba mais: [Estilo Global](#)

## Dimensionar a Área

Mudar o tamanho da área (canvas) vai escalar todos os seus conteúdos (textos, imagens, componentes, etc.) e slides.

```

padrão: 980
como a área fica menor, o tamanho visual vai ficar maior
canvasWidth: 800

```

## Usar o Transform

Nós disponibilizamos um componente integrado chamado `<Transform />`, que é uma pequena alternativa que engloba a propriedade transform do CSS.

```
<Transform :scale="1.4">
- Item 1
- Item 2
</Transform>
```

# Hospedagem Estática

## Compilando Single Page Applications (SPA)

Você também pode compilar os slides em uma aplicação SPA auto-hospedável:

```
$ slidev build
```

A aplicação gerada estará disponível na pasta `dist/` e a partir daí você pode hospedá-la no [GitHub Pages](#), [Netlify](#), [Vercel](#), ou em qualquer outra plataforma que preferir. Agora você pode compartilhar seus slides com o todo mundo com um único link.

### Diretório Base

Para implantar seus slides em sub-rotas, você precisará passar a opção `--base`. Por exemplo:

```
$ slidev build --base /palestras/minha-palestra-legal/
```

Consulte a [documentação do Vite](#) para mais detalhes.

## Disponibilize um PDF para download

Você pode disponibilizar um PDF para download aos visitantes da sua aplicação SPA com a seguinte configuração:

```

 download: true

```

O Slidev vai gerar um arquivo PDF na compilação e um botão de download aparecerá na aplicação SPA.

Você também pode prover uma URL customizada para o PDF. Nesse caso, o processo de renderização do arquivo será pulado.

```

 download: 'https://myside.com/my-talk.pdf'

```

## Exemplos

Aqui estão alguns exemplos de aplicações SPA exportadas:

- [Starter Template](#)
- [Composable Vue por Anthony Fu](#)

Para mais, veja a [Galeria](#).

# Hospedando

Nós recomendamos usar o comando `npm init slidev@lastest` para iniciar seu projeto, que já contém as configurações necessárias para serviços de hospedagem.

## Netlify

- [Netlify](#)

Crie o arquivo `netlify.toml` na raiz do projeto com o seguinte conteúdo.

```
[build.environment]
 NODE_VERSION = "14"

[build]
 publish = "dist"
 command = "npm run build"

[[redirects]]
 from = "/"
 to = "/index.html"
 status = 200
```

Depois vá para o seu painel de controle do Netlify e crie um novo site com o repositório.

## Vercel

- [Vercel](#)

Crie o arquivo `vercel.json` na raiz do projeto com o seguinte conteúdo.

```
{
 "rewrites": [
 { "source": "/(.*)", "destination": "/index.html" }
]
}
```

Depois vá para o seu painel de controle da Vercel e crie um novo site com o repositório.

## GitHub Pages

- [GitHub Pages](#)

Crie o arquivo `.github/workflows/deploy.yml` com o seguinte conteúdo para implantar seus slides no Github Pages pelo Github Actions.

```
name: Deploy pages
on: push
jobs:
 deploy:
 runs-on: ubuntu-latest
 steps:
 - uses: actions/checkout@v2
 - uses: actions/setup-node@v2
```

```
with:
 node-version: '14'
- name: Install dependencies
 run: npm install
- name: Build
 run: npm run build
- name: Deploy pages
 uses: crazy-max/ghaction-github-pages@v2
 with:
 build_dir: dist
env:
 GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
```

# Primeiros Passos

## Introdução

Slidev (slide + dev, `/slайдив/`) é uma ferramenta baseada em tecnologias web para criar e apresentar slides. O Slidev foi projetado para desenvolvedores focarem em escrever conteúdo em Markdown e ainda ter o poder do HTML e dos componentes do Vue para fornecer layouts e designs perfeitos com demonstrações interativas de suas apresentações.

É usado um arquivo Markdown cheio de recursos para gerar slides bonitos com uma experiência de *refresh* instantâneo, além de várias integrações embutidas como desenvolvimento de código em tempo real, exportação pra PDF, gravação da apresentação, e assim por diante. Como tem como base tecnologias da web, você pode fazer tudo com o Slidev - as possibilidades são infinitas.

Você pode saber mais sobre a lógica por trás do projeto na seção [Por que Slidev](#).

## Funcionalidades

-  **Baseado em markdown** - use seus editores fluxos de trabalho favoritos
-  **Feito para desenvolvedores** - destaque de sintaxe integrado, desenvolvimento de código em tempo real, etc.
-  **Suporte a Temas** - temas podem ser compartilhados e usados com pacotes do npm
-  **Estiloso** - utilidades do Windi CSS sob demanda, fácil incorporação de CSS
-  **Interativo** - suporta perfeitamente a incorporação de componentes do Vue
-  **Modo Apresentador** - use outra janela, ou até mesmo seu celular para controlar seus slides
-  **Desenhos** - desenhe e anote em seus slides
-  **LaTeX** - suporte integrado à equações matemáticas em LaTeX
-  **Diagramas** - cria diagramas com descrições em texto
-  **Ícones** - acesse ícones de qualquer conjunto diretamente
-  **Editores** - editor integrado, ou uma [extensão para o VS Code](#)
-  **Gravação** - gravação integrada e visão da câmera
-  **Portátil** - exporte para PDF, PNGs, ou até mesmo uma aplicação SPA hospedável
-  **Rápido** - *refresh* instantâneo provido pelo Vite
-  **Customizável** - usando plug-ins do Vite, componentes do Vue, ou qualquer pacote do npm

## Tecnologias

O Slidev é possível pela combinação dessas ferramentas e tecnologias.

- [Vite](#) - Uma ferramenta para front-end extremamente rápida
- [Vue 3 com Markdown](#) - Foco no conteúdo a ainda ter o poder do HTML e dos componentes do Vue quando precisar
- [Windi CSS](#) - Framework CSS com utilidades sob demanda, estilize seus slides com facilidade
- [Prism, Shiki, Monaco Editor](#) - Suporte a blocos de código como codificação em tempo real
- [RecordRTC](#) - Gravação e visão da câmera integrados

- Família [VueUse](#) - `@vueuse/core`, `@vueuse/head`, `@vueuse/motion`, etc.
- [Iconify](#) - Coleção de ícones.
- [Draau](#) - Suporte à desenhos e anotações
- [KaTeX](#) - Renderização do LaTeX.
- [Mermaid](#) - Diagramas baseados em texto.

## Criando Sua Primeira Apresentação

### Experimente online

[sli.dev/new](https://sli.dev/new)



[Open in StackBlitz](#)

### Crie Localmente

Com NPM:

```
$ npm init slidev
```

Com Yarn:

```
$ yarn create slidev
```

Siga as instruções e comece a fazer seus slides agora mesmo! Para mais detalhes sobre a sintaxe do markdown, dê uma olhada no [guia de sintaxe](#).

## Interface de Linha de Comando

Em um projeto onde o Slidev está instalado, você pode usar o binário `slidev` nos seus scripts do npm.

```
{
 "scripts": {
 "dev": "slidev", // inicia um servidor para desenvolvimento
 "build": "slidev build", // compila a aplicação SPA para produção
 "export": "slidev export" // exporta seus slides para pdf
 }
}
```

Você também pode usá-lo com o `npx`

```
$ npx slidev
```

Rode `slidev --help` para mais opções disponíveis.

## Sintaxe do Markdown

O Slidev lê o seu arquivo `slides.md` na raiz do seu projeto e o converte em slides. A qualquer alteração que você faça nele, o conteúdo dos slides serão atualizados imediatamente. Por exemplo:

```
Slidev

Olá Mundo

Página 2

Use blocos de código diretamente para o destaque

//```ts
console.log('Olá, Mundo!')
//```

Página 3
```

Leia mais sobre a sintaxe Markdown do Slidev no [guia de sintaxe](#).

# Instalação

## Modelo Inicial

Slidev requer **Node.js >=14.0**

O melhor jeito de começar é usando nosso modelo oficial.

Com NPM:

```
$ npm init slidev@latest
```

Com Yarn:

```
$ yarn create slidev
```

Siga as instruções e vai abrir automaticamente pra você uma apresentação de slides em <http://localhost:3030/>.

A apresentação também contém a configuração básica e uma breve demonstração de como começar com o Slidev.

## Instalação Manual

Se ainda assim preferir instalar o Slidev manualmente ou se quiser integrá-lo em seus projetos existentes, você pode fazer assim:

```
$ npm install @slidev/cli @slidev/theme-default
```

```
$ touch slides.md
```

```
$ npx slidev
```

Observe que se você estiver usando o [pnpm](#), será necessário habilitar a opção [shamefully-hoist](#) para que o Slidev funcione corretamente:

```
echo 'shamefully-flatten=true' >> .npmrc
```

# Instalação Global

Disponível a partir da v0.14

Você pode instalar o Slidev globalmente com o seguinte comando

```
$ npm i -g @slidev/cli
```

E agora você pode usar comando `slidev` em qualquer lugar sem ter que criar um projeto toda hora.

```
$ slidev
```

Este comando também vai tentar o `@slidev/cli` local se ele for encontrado nos `node_modules`.

## Instalação no Docker

Se você precisar de uma maneira rápida de rodar uma apresentação com container, você pode usar a imagem do `docker` mantida por [stig124](#), ou construir sua própria.

Consulte o [repositório do slidevjs/container](#) para mais detalhes.

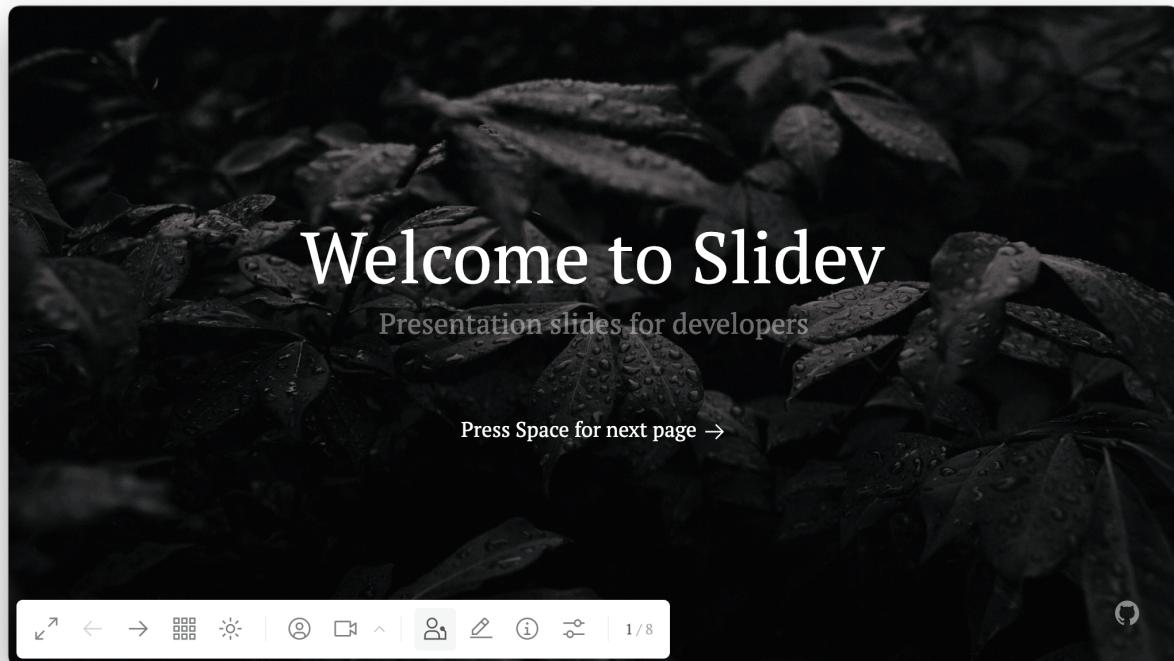
---

[Go to TOC](#)

# Navegação

## Barra de Navegação

Mova seu mouse para o canto inferior esquerdo da página do Slidev, a barra de navegação aparecerá.

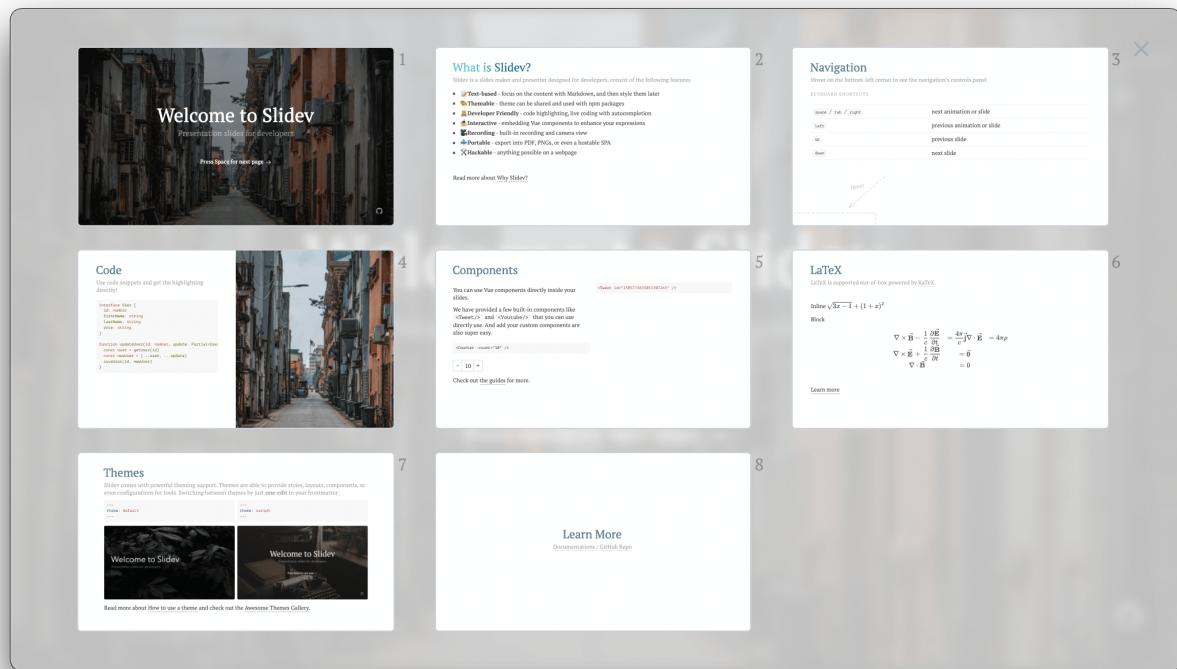


Atalhos	Botão	Descrição
f		alterna a tela cheia
direita / espaço		próxima animação ou slide
esquerda		animação ou slide anterior
cima	-	slide anterior
baixo	-	próximo slide
o		alterna a <a href="#">visualização dos slides</a>
d		alterna o modo escuro
-		alterna a <a href="#">visão da câmera</a>
-		<a href="#">gravação</a>
-		<a href="#">entra no modo apresentador</a>
-		<a href="#">alterna o editor integrado</a>

Atalhos	Botão	Descrição
-		faz download dos slides (só aparece na <a href="#">compilação SPA</a> )
-		mostra informações sobre os slides
-		mostra o menu de configurações
g	-	mostra o ir para...

## Visualização dos Slides

Ao apertar o ou clicar no botão na barra de navegação, você pode ter uma visão geral dos slides para que você possa alterná-los facilmente.



[Go to TOC](#)

# Modo Apresentador

Clique no botão no painel de navegação, ou visite manualmente <http://localhost:3030/presenter> para entrar no modo apresentador. Sempre que você entrar no modo de apresentador, outras instâncias da página ficarão automaticamente sincronizadas com o apresentador.

The screenshot shows a presentation slide titled "Welcome to Sliddev" with the subtitle "Presentation slides for developers". A note at the bottom says "Press Space for next page →". To the right of the slide, there is a sidebar with the title "What is Sliddev?" and a list of features:

- **Dev based**: Focus on the content and mechanics, and their style sheets last
- **Mobile friendly**: Works on mobile devices and tablets with great performance
- **Developer friendly**: Code highlighting, live coding with autocompletion
- **Collaboration**: Built-in commenting and discussion for presentations
- **Recording**: Built-in recording and camera view
- **Export**: Export slide PDF, PPTX, or even a executable IPA
- **Shareable**: Getting people involved is a breeze!

Below the sidebar, a note states: "The last comment block of each slide will be treated as slide notes. It will be visible and editable in Presenter Mode along with the slide. [Read more in the docs](#)".

At the bottom of the slide, there is a navigation bar with icons for back, forward, search, and other presentation controls, and a status bar showing "1 / 7".

---

[Go to TOC](#)

# Gravando

O Slidev tem gravação e visão da câmera integrados. Você pode usá-las para gravar sua apresentação facilmente em um lugar.

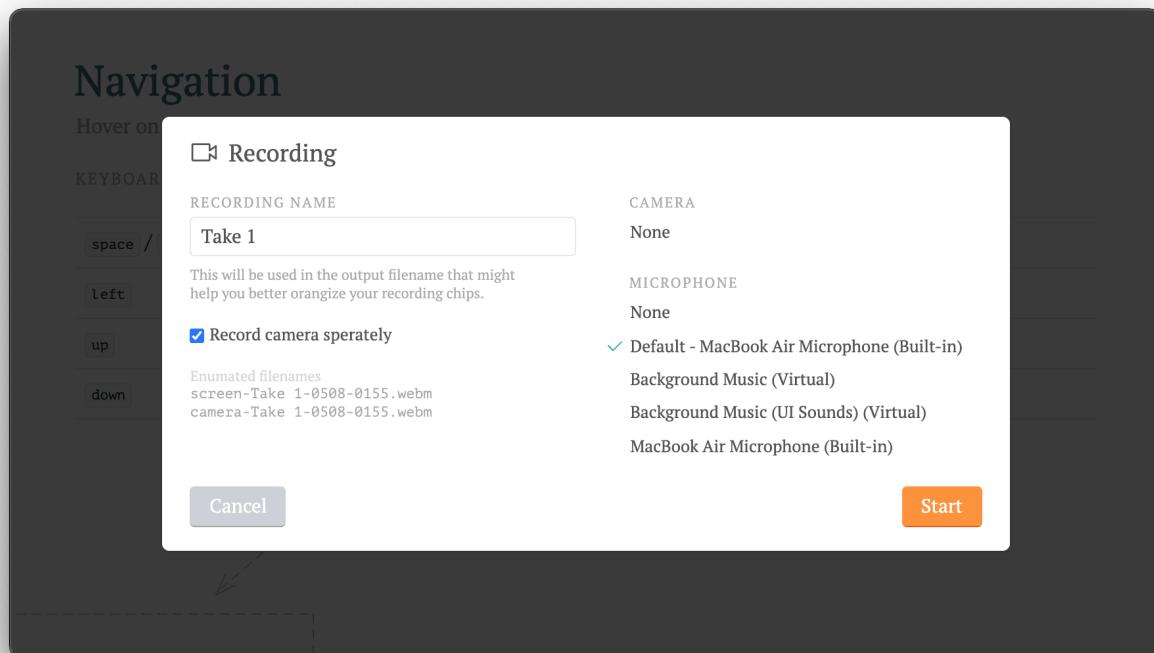
## Visão da Câmera

Clique no botão no painel de navegação para mostrar sua câmera na apresentação. Você pode arrastar para movê-la, e usar o manipulador no canto inferior direito para redimensioná-la. O tamanho e a posição vai continuar no `localStorage`, por isso vai permanecer mesmo após recarregamentos, então você não precisa se preocupar com isso.

## Gravando

Clicar no botão no painel de navegação vai abrir uma caixa de diálogo pra você. Lá você pode escolher entre gravar sua câmera embutida nos slides ou separar as gravações em dois arquivos.

Esta funcionalidade é possível graças ao [RecordRTC](#) e usa o [WebRTC API](#).




---

[Go to TOC](#)

# Sintaxe do Markdown

Os slides são escritos em **um único arquivo markdown** (por padrão `./slides.md`).

Você pode usar [os recursos do Markdown](#) normalmente, com suporte adicional de HTML *inline* e Componentes do Vue. Estilização usando o [Windi CSS](#) também é suportada. Use `---` seguido de uma nova linha para separar seus slides.

## # Slides

Olá, Mundo!

---

## # Página 2

Use blocos de código diretamente para ter destaque na sintaxe

```
//```ts
console.log('Olá, Mundo!')
//``
```

---

## # Página 3

Você pode usar o Windi CSS e componentes Vue diretamente para estilizar e enriquecer seus slides.

```
<div class="p-3">
 <Tweet id="20" />
</div>
```

# Front Matter & Layouts

Especifique layouts e outros metadados para cada slide convertendo os separadores em [blocos front matter](#). Cada frontmatter começa e termina com três traços (`---`). Os textos entre os traços são dados no formato [YAML](#). Por exemplo:

---

`layout: cover`

---

## # Slides

Esta é a página capa.

---

`layout: center`  
`background: './images/background-1.png'`  
`class: 'text-white'`

---

## # Página 2

Esta é uma página com o layout `center` e uma imagem de fundo.

---

# Página 3

Está pe uma página padrão sem nenhum metadado adicional.

Dê uma olhada em [customização](#) para mais detalhes.

## Blocos de Código

Um grande motivo pelo qual eu estou desenvolvendo o Slidev é a necessidade de fazer meu código aparecer perfeitamente nos slides. Então como você deve estar pensando, você pode usar blocos de código no estilo do Markdown para destacar seu código.

```
//``ts
console.log('Olá, Mundo!')
//``
```

Nós suportamos o [Prism](#) e o [Shiki](#) como destacadores de sintaxe. Consulte [a seção de destacadores](#) para mais detalhes.

## Realce de Linha

Para realçar linhas específicas, simplesmente adicione o número das linhas em chaves `{}`. A contagem do número das linhas começa a partir do 1.

```
//``ts {2,3}
function add(
 a: Ref<number> | number,
 b: Ref<number> | number
) {
 return computed(() => unref(a) + unref(b))
}
//``
```

Para definir os realces em vários passos, você pode usar `|` para separá-los. Por exemplo:

```
//``ts {2-3|5|all}
function add(
 a: Ref<number> | number,
 b: Ref<number> | number
) {
 return computed(() => unref(a) + unref(b))
}
//``
```

Isto vai primeiro realçar `a: Ref<number> | number` e `b: Ref<number> | number`, aí `return computed(() => unref(a) + unref(b))` após um clique, e por último, o bloco inteiro. Aprenda mais no [guias de animações no clique](#)

## Editor Monaco

Sempre que quiser fazer alguma modificação na apresentação, simplesmente adicione `{monaco}` após o id da linguagem — isto transforma o bloco em um editor Monaco com todos os seus recursos!

```
//``ts {monaco}
console.log('OláMundo')
//``
```

Aprenda mais sobre [configurar Monaco](#).

## Estilos Embutidos

Você pode usar a tag `<style>` diretamente no seu Markdown para sobrepor estilos no **slide atual**.

```
Isso está Vermelho

<style>
h1 {
 color: red
}
</style>

O próximo slide não é afetado
```

A tag `<style>` no Markdown sempre segue um [escopo](#). Para redefinir estilos globalmente, dá uma olhada na [seção de customização](#)

Graças ao [Windi CSS](#), você pode usar aninhamento de CSS diretamente, além de diretrizes (e.g. `@apply`).

```
Slides

> Olá `mundo`

<style>
blockquote {
 code {
 @apply text-teal-500 dark:text-teal-400;
 }
}
</style>
```

## Recursos Estáticos

Assim como você faria no markdown, você pode usar imagens apontando para uma URL remota ou local.

Para recursos estáticos, o `vite-plugin-remote-assets` embutido vai criar um cache na memória na primeira execução para que você possa ter o carregamento instantâneo depois, até pra imagens maiores.

```
![Imagen Remota](https://sli.dev/favicon.png)
```

Para recursos locais, os coloque na pasta `public` e depois a referencie com uma **barra no início**.

```
! [Imagen Local](foto.png)
```

Se quiser aplicar tamanhos ou estilos customizados, você pode converter para uma tag `<img>`

```

```

## Notas

Você também pode fazer anotações em cada slide. Elas aparecerão no [Modo Apresentador](#) para que você possa usá-las como referência durante apresentações.

Em Markdown, o último bloco de comentário em cada slide será considerado uma nota.

```

layout: cover

Página 1

Esta é a página de capa.

<!-- Isto é uma nota -->

Página 2

<!-- Isto NÃO é uma nota porque vem antes do conteúdo do slide -->

A segunda página

<!--
Isto é outra nota
-->
```

## Ícones

O Slidev te permite acessar quase todos os conjuntos de ícones open-source **diretamente** no seu markdown. Graças ao `vite-plugin-icons` e ao [Iconify](#).

Os nomes seguem a convenção do [Iconify](#) `{nome-da-coleção}-{nome-do-ícone}`. Por exemplo:

- `<mdi-account-circle />` - do [Material Design Icons](#)
- `<carbon-badge />` - do [Carbon](#)
- `<uim-rocket />` - do [Unicons Monochrome](#)
- `<twemoji-cat-with-tears-of-joy />` - do [Twemoji](#)
- `<logos-vue />` - do [SVG Logos](#)
- E muito mais...

Navegue e pesquise por todos os ícones disponíveis com o [Ícones](#).

## Estilizando Ícones

Você pode estilizar os ícones como qualquer outro elemento HTML. Por exemplo:

```
<uim-rocket />
<uim-rocket class="text-3xl text-red-400 mx-2" />
<uim-rocket class="text-3xl text-orange-400 animate-ping" />
```

## Slots

Disponível desde a v0.18

Alguns layouts podem fornecer vários pontos de contribuição usando [slots nomeados do Vue](#).

Por exemplo, no `layout two-cols`, você pode ter duas colunas lado a lado: a da esquerda (slot `default`) e a da direita (slot `right`).

```

layout: two-cols

<template v-slot:default>
Esquerda
Isto é mostrado à esquerda

</template>
<template v-slot:right>
Direita
Isto é mostrado à direita
</template>
```

### Esquerda

Isto é mostrado à esquerda

### Direita

Isto é mostrado à direita

Nós também fornecemos uma sintaxe abreviada `::nome::` para o nome do slot. A seguinte exemplo funciona exatamente como o anterior.

```

layout: two-cols

Esquerda
```

```
Isto é mostrado à esquerda
```

```
::right::
```

```
Direita
```

```
Isto é mostrado à direita
```

Você também pode explicitamente especificar o slot padrão para ter uma ordenação customizada:

```

```

```
layout: two-cols
```

```

```

```
::right::
```

```
Direita
```

```
Isto é mostrado à direita
```

```
::default::
```

```
Esquerda
```

```
Isto é mostrado à esquerda
```

## Configurações

Todas as configurações necessárias podem ser definidas no arquivo Markdown. Por exemplo:

```

```

```
theme: serif
```

```
layout: cover
```

```
background: 'https://source.unsplash.com/1600x900/?nature,water'
```

```

```

```
Slidel
```

```
Esta a página de capa.
```

Aprenda mais sobre as [configurações do frontmatter](#)

## LaTeX

O Slidev vem com suporte a LaTeX incluído, usando o [KaTeX](#) para isso.

### Em Linha

Insira um `$` em cada lado do seu LaTeX para renderizá-lo na mesma linha.

```
$\sqrt{3x-1}+(1+x)^2$
```

### Bloco

Use dois (`$$`) para renderização em bloco. Este modo mostra símbolos maiores e centraliza o resultado.

```
$$
\begin{array}{c}
\nabla \times \vec{\mathbf{B}} - \frac{1}{c} \nabla \frac{\partial \vec{\mathbf{E}}}{\partial t} + \frac{4\pi}{c} \vec{\mathbf{c}} \cdot \vec{\mathbf{E}} = 4\pi \rho \\
\\
\nabla \times \vec{\mathbf{E}} + \frac{1}{c} \nabla \frac{\partial \vec{\mathbf{B}}}{\partial t} = \vec{\mathbf{0}} \\
\\
\nabla \cdot \vec{\mathbf{B}} = 0 \\
\\
\end{array}
$$
```

Aprenda mais: [Demonstração](#) | [KaTeX](#) | [markdown-it-katex](#)

## Diagramas

Você também pode criar diagramas / gráficos a partir de descrições em texto no seu Markdown, renderizado pelo [Mermaid](#).

Blocos de código marcados como `mermaid` serão convertidos para diagramas, por exemplo:

```
//```mermaid
sequenceDiagram
 Valentina->Enzo: Olá Enzo, tudo bem?
 Note over Valentina,Enzo: Uma interação clássica
//````
```

Você pode até passar um objeto com opções para especificar o tamanho e o tema. A sintaxe do objeto é de um objeto literal do JavaScript, você vai precisar adicionar aspas (`'`) para textos e usar vírgula (`,`) entre cada chave (`theme`, `scale`, etc).

```
//```mermaid {theme: 'neutral', scale: 0.8}
graph TD
 B[Texto] --> C{Decisão}
 C -->|Uml| D[Resultado 1]
 C -->|Dois!| E[Resultado 2]
//````
```

Aprenda mais: [Demonstração](#) | [Mermaid](#)

## Múltiplas Entradas

Disponível desde a v0.15

Você pode dividir seu arquivo `slides.md` em múltiplos arquivos e organizá-los como preferir.

`slides.md` :

## # Página 1

Está é uma página comum

```

src: ./subpagina2.md

```

<!-- esta página será carregada a partir do arquivo './subpagina2.md' -->  
Conteúdo na linha será ignorado

`subpage2.md :`

## # Página 2

Esta página é de outro arquivo

**Mesclagem de Frontmatter**

Você pode prover frontmatters tanto para sua entrada principal quanto em páginas markdown externas. Caso elas tenham chaves repetidas, as da **entrada principal têm maior prioridade**. Por exemplo:

`slides.md :`

```

src: ./cover.md
background: https://sli.dev/bar.png
class: text-center

```

`cover.md :`

```

layout: cover
background: https://sli.dev/foo.png

```

## # Capa

Página de Capa

Vai acabar sendo equivalente à seguinte página:

```

layout: cover
background: https://sli.dev/bar.png
class: text-center

```

## # Capa

Página de Capa

**Reutilização de Página**

Com o suporte a múltiplas entradas, reutilizar páginas pode ser simples. Por exemplo:

```

src: ./capa.md

src: ./intro.md

src: ./conteudo.md

reutilização
src: ./conteudo.md

```

# Por que o Slidev

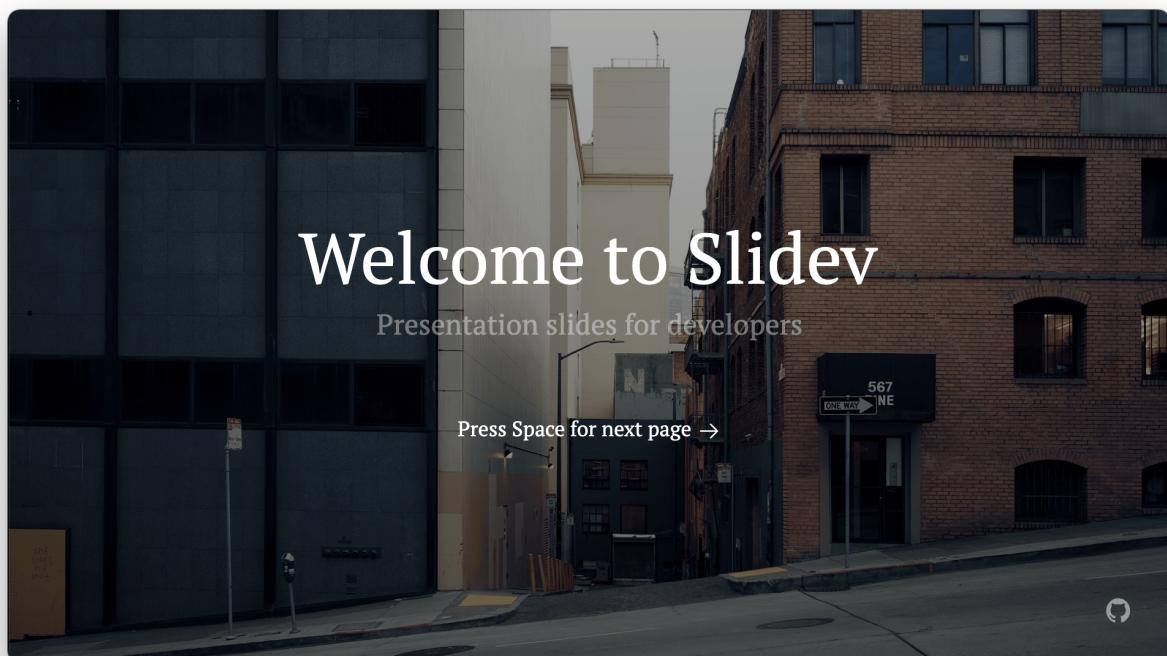
Existem inúmeras ferramentas para criar slides cheias de recurso, de uso geral, como [Microsoft PowerPoint](#) e [Apple Keynote](#). Elas funcionam muito bem para fazer slides legais com animações, gráficos, e muitas outras coisas, além de serem bem intuitivas e fáceis de aprender. Então por que gastar tempo fazendo o Slidev?

O Slidev visa fornecer a flexibilidade e a interatividade para desenvolvedores fazerem suas apresentações ainda mais interessantes, expressivas, e atraentes usando ferramentas e tecnologias das quais já estão familiarizados.

Ao trabalhar com editores **WYSIWYG**, é bem fácil se distrair com as opções de estilização. O Slidev corrige isso separando o conteúdo dos recursos visuais. Isto te permite focar em uma coisa de cada vez, além de reutilizar temas da comunidade. O Slidev não pretende substituir completamente outras ferramentas de slides. Em vez disso, se concentra em atender à comunidade de desenvolvedores.

Nota: **WYSIWYG** é um acrônimo para "What You See Is What You Get", que seria algo como "O que você vê é o que você obtém".

## Slidev



Aqui estão alguns dos recursos mais legais do Slidev:

## Baseado em markdown

O Slidev usa um formato estendido do Markdown para armazenar seus slides em um único arquivo de texto. Isto te permite focar em produzir o conteúdo. E já que o conteúdo é separado dos estilos, isto também te permite trocar de temas sem esforço.

Aprenda mais sobre [a sintaxe Markdown do Slidev](#).

## Suporte a Temas

Temas para o Slidev podem ser compartilhados e instalados usando pacotes do npm. Você pode aplicá-los com uma linha de configuração.

Dê uma olhada na [galeria de temas](#) ou [aprenda a escrever um tema](#).

## Feito para Desenvolvedores

O Slidev fornece suporte de primeira à blocos de código. Suporta tanto o [Prism](#) quanto o [Shiki](#) para conseguir um destaque perfeito da sintaxe, ainda tendo a possibilidade de modificação do código a qualquer hora. Com o [editor Monaco](#) embutido, ele também te permite escrever código em tempo real / demonstrá-lo na sua apresentação com *autocomplete*, tipo da variável ao passar o mouse em cima, e até suporte à checagem de tipo do TypeScript.

Aprenda mais sobre [destacadores](#) e [configurações do Monaco](#).

## Rápido

O Slidev é construído com [Vite](#), [Vue 3](#) e [Windi CSS](#), o que te proporciona uma experiência espetacular de desenvolvimento. Toda mudança que você fizer refletirá **instantaneamente** em seus slides.

Saiba mais sobre [nossas tecnologias](#)

## Interativo & Expressivo

Você pode desenvolver componentes do Vue customizados e usá-los diretamente no seu arquivo markdown. Você também pode interagir com eles nas apresentações para expressar sua ideia de uma forma mais intuitiva e interessante.

## Suporte a Gravação

O Slidev fornece gravação e visão da câmera integrados. Você pode compartilhar a sua apresentação com a visão da sua câmera embutida, ou gravar e salvar a sua tela e câmera separadamente. Tudo de uma vez só, sem a necessidade de nenhuma ferramenta adicional.

Aprenda mais sobre [gravação aqui](#).

## Portátil

Exporte seus slides para PDF, PNGs, ou até mesmo uma aplicação SPA hospedável com um único comando, e os compartilhe em qualquer lugar.

Leia mais sobre isto na [documentação de exportação](#).

## Customizável

Sendo baseado em tecnologias web, qualquer coisa que pode ser feita numa aplicação web também é possível com o Slidev. Por exemplo, WebGL, requisição à APIs, iframes, ou até mesmo compartilhamento ao vivo. Só depende da sua imaginação!

## De uma Chance

Brincar um pouco com o Slidev vai te dizer mais do que mil palavras. E está a apenas um comando de você:

```
$ npm init slidev
```

Ou tenha uma breve prévia:

Slidev First Preview Demo

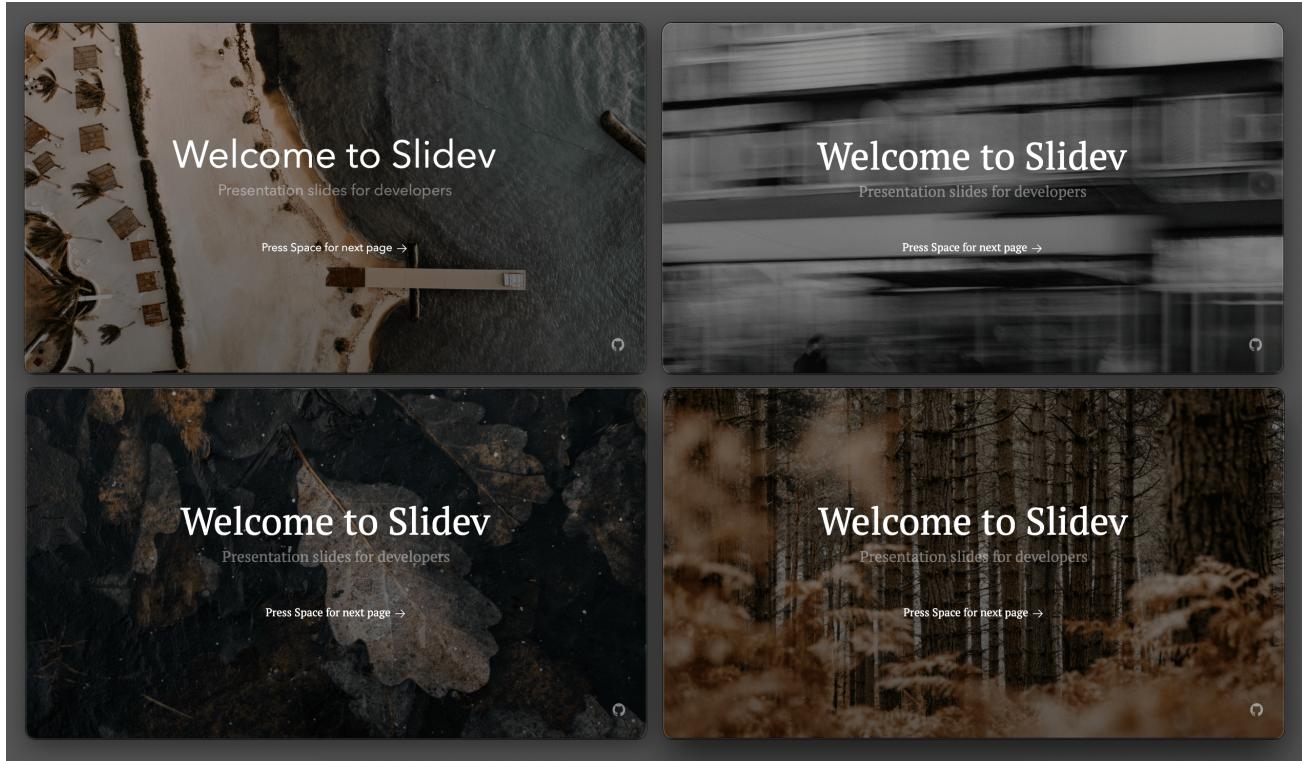


---

[Go to TOC](#)

# Capas Selecionadas

Nós selecionamos algumas imagens de capa para demonstrar nosso modelo inicial.



```

imagem aleatória da coleção
background: https://source.unsplash.com/collection/94734566/1920x1080

```

Se você gostar de qualquer uma delas, dê uma olhada na nossa [coleção no Unsplash](#) e descubra os autores.

---

[Go to TOC](#)

# Recursos de Aprendizagem

## English

### Videos

Slidev - one of the best presentation software and it is free!



### Articles

- [Tips To Turn R Markdown Into Slidev Presentation](#) by Hiroaki Yutani

## 中文

- [Slidev : 一个用Markdown写slides的神器](#) by 梦里风林
- [神器！这款开源项目可以让你使用 Markdown 来做 PPT！](#) by [Github掘金计划](#)
- [【用 markdown 写 Slide!】神器 Slidev 的安装及 bug 解决](#) by HaloHoohoo

## 日本语

- [開発者のためのスライド作成ツール Slidev がすごい](#) by ryo\_kawamata
- [Markdownでオシャレなスライドを作るSli.dev](#) by Nobuko YAMADA

---

[Go to TOC](#)

# Galeria

Palestras / Apresentações usando Sliddev.

---

[Go to TOC](#)

# Galeria de Temas

Navegue por temas incríveis disponíveis para o Slidev aqui.

Leia mais sobre [como usar um tema](#), ou [como escrever o seu próprio](#) e compartilhar com a comunidade!

## Temas Oficiais

## Temas da Comunidade

Aqui estão os temas selecionados feitos pela comunidade.

## Mais Temas

Ache todos os [temas disponíveis no NPM](#).

---

[Go to TOC](#)

# Usar um Tema

Mudar o tema no Slidev é surpreendentemente fácil. Tudo que você precisa para fazer isso é adicionar o campo `theme:` no seu frontmatter.

```

theme: serif

```

Você pode iniciar o servidor, que vai te solicitará instalar o tema automaticamente.

? The theme "@slidev/theme-seriph" was not found in your project, do you want to install it now? › (Y/n)

ou instale o tema manualmente via

```
$ npm install @slidev/theme-seriph
```

E é só isso, aproveite o novo tema! Para mais detalhes sobre o uso, você pode consultar o README do tema.

Quer compartilhar seu tema? Aprenda sobre [como escrever um tema](#).

## Ejetar um Tema

Se você quiser ter controle total do tema atual, você pode **ejetá-lo** para os seus arquivos locais e modificá-lo como quiser. Rodando o seguinte comando:

```
$ slidev theme eject
```

Isto ejetará o tema que você estiver usando para `./theme`, e vai alterar o seu frontmatter para:

```

theme: ./theme

```

Isto pode ser útil se você quiser fazer um tema baseado em um existente. Se fizer isso, lembre-se de mencionar o autor original do tema :)

## Tema Local

Como você provavelmente viu na seção anterior, você pode ter um tema local para os seus projetos. Tendo um **caminho relativo** no campo `theme`.

```

theme: ./caminho/para/o/tema

```

Consulte [como escrever um tema](#) para mais detalhes.

---

[Go to TOC](#)

# Escrever um Tema

Pra começar, nós te recomendamos usar nosso gerador como base para o seu primeiro tema:

```
$ npm init slidev-theme
```

Depois você pode modificar e brincar com o tema. Você também pode consultar os [temas oficiais](#) como exemplos.

## Capacidade

Um tema pode contribuir nos seguintes pontos:

- Estilos globais
- Prover configurações padrão (fontes, esquema de cor, destacadores, etc.)
- Prover layouts customizados ou sobrepor um existentes
- Prover componentes customizados ou sobrepor existentes
- Estender as configurações do Windi CSS
- Configurar ferramentas como Monaco e Prism

## Convenções

Temas são publicados no registro do npm, e eles devem seguir as seguintes convenções:

- Nome do pacote deve começar com `slidev-theme-`, por exemplo: `slidev-theme-incrivel`
- Adicione `slidev-theme` e `slidev` no campo `keywords` do seu `package.json`

## Configuração

Para configurar o ambiente de teste para o seu tema, você pode criar um arquivo `example.md` com o seguinte frontmatter, para dizer ao Slidev que você está usando o diretório atual como um tema.

```

theme: ./

```

Opcionalmente, você também pode adicionar alguns scripts ao seu `package.json`

```
// package.json
{
 "scripts": {
 "dev": "slidev example.md",
 "build": "slidev build example.md",
 "export": "slidev export example.md",
 "screenshot": "slidev export example.md --format png"
 }
}
```

Para publicar seu tema, simplesmente execute `npm publish` e pronto. Não há nenhum processo de compilação necessário (o que significa que você pode publicar arquivos `.vue` e `.ts` diretamente, o Slidev é esperto o bastante para entendê-los).

Os pontos de contribuição do tema seguem as mesmas convenções das customizações locais, consulte [a documentação das convenções de nome](#).

## Configurações Padrão

Disponível desde a v0.19

Um tema pode prover [configurações padrão](#) por meio do `package.json`.

```
// package.json
{
 "slidev": {
 "default": {
 "aspectRatio": "16/9",
 "canvasWidth": 980,
 "fonts": {
 "sans": "Robot",
 "mono": "Fira Code"
 }
 }
 }
}
```

Fontes serão automaticamente importadas do [Google Fonts](#).

Aprenda mais sobre [fontes](#) e [configurações do frontmatter](#)

## Metadados do Tema

### Esquema de Cores

Por padrão, o Slidev assume que os temas suportam tanto o modo claro quanto o escuro. Se você quiser que seu tema seja apresentado em apenas um esquema de cores, você vai precisar especificar isto explicitamente no `package.json`:

```
// package.json
{
 "name": "slidev-theme-meu-tema-legal",
 "keywords": [
 "slidev-theme",
 "slidev"
],
 "slidev": {
 "colorSchema": "light" // ou "dark" ou "both"
 }
}
```

Para acessar o modo escuro quando estiver criando os estilos do seu tema, você pode usar a especificação de modo escuro do CSS com a classe `dark`:

```
/* css generalista aqui */

html:not(.dark) {
 /* css para modo claro aqui */
}

html.dark {
 /* css para modo escuro aqui */
}
```

O Slidev alterna a classe `dark` no elemento `html` da página para trocar o esquema de cores.

## Destacador

As cores do destacador de sintaxe também são provenientes do tema. Nós suportamos tanto o [Prism](#) quanto o [Shiki](#). Para mais informação consulte a [documentação do destaque de sintaxe](#).

Você pode suportar um deles, ou os dois. Consulte os exemplos de configuração padrão de tema `./styles/prism.css` / `./setup/shiki.ts`.

Além disso, lembre-se de especificar os destacadores suportados no seu `package.json`:

```
// package.json
{
 "slidev": {
 "highlighter": "shiki" // ou "prism" ou "all"
 }
}
```

## Versão do Slidev

Se o tema estiver usando um recurso específico do Slidev que foi introduzido recentemente, você pode definir a versão mínima do Slidev necessária para ter o seu tema funcionando bem:

```
// package.json
{
 "engines": {
 "slidev": ">=0.19.3"
 }
}
```

Se usuários estiverem usando versões mais antigas do Slidev, será gerado um erro.

# Colophon

This book is created by using the following sources:

- Slidev - Português-BR
- GitHub source: [slidevjs/docs-br](https://github.com/slidesjs/docs-br)
- Created: 2022-11-27
- Bash v5.2.2
- Vivliostyle, <https://vivliostyle.org/>
- By: @shinokada
- GitHub repo: <https://github.com/shinokada/markdown-docs-as-pdf>