

SLIDEV Docs - Español



Sliddev

Table of contents

• Es - README	4
• Es - TRANSLATIONS	5
• Builtin - Components	8
• Builtin - Layouts	10
• Custom - Config katex	14
• Custom - Config mermaid	15
• Custom - Config monaco	16
• Custom - Config shortcuts	18
• Custom - Config vite	19
• Custom - Config vue	21
• Custom - Config windicss	22
• Custom - Directory structure	23
• Custom - Fonts	27
• Custom - Global layers	30
• Custom - Highlighters	32
• Custom - Index	34
• Custom - Vue context	36
• Guide - Animations	38
• Guide - Drawing	42
• Guide - Editors	44
• Guide - Exporting	47
• Guide - Faq	49
• Guide - Hosting	52
• Guide - Index	55
• Guide - Install	58
• Guide - Navigation	61
• Guide - Presenter mode	63
• Guide - Recording	64
• Guide - Syntax	65
• Guide - Why	74
• Resources - Covers	77
• Resources - Learning	78
• Es - Showcases	79
• Themes - Gallery	80

- Themes - Use _____ 81
- Themes - Write a /theme _____ 83

sli.dev

Spanish documentation for [Slidev](#)

Translations

	Repo	Site	Maintainers
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
日本語	docs-ja	ja.sli.dev	@IkumaTadokoro

Start Server Locally

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

And then visit <http://localhost:3000>

Or install the [Vite extension for VS Code](#) to edit side-by-side.

Help on Translating

See [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/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
- `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 integrados

La documentación de esta sección está todavía en progreso. Mientras tanto, puedes echar un vistazo al [código fuente](#) directamente.

Toc

Inserta una tabla de contenidos (TOC).

Si deseas que una diapositiva no aparezca en el componente `<Toc>`, puedes usar el siguiente frontmatter:

```
---  
hideInToc: true  
---
```

Los títulos se muestran con el componente `<Titles>`

Uso

```
<Toc />
```

Parámetros

- `columns` (`string | number` , por defecto: `1`) El número de columnas a mostrar
- `maxDepth` (`string | number` , por defecto: `Infinity`): La profundidad máxima a mostrar
- `minDepth` (`string | number` , por defecto: `1`): La profundidad mínima a mostrar
- `mode` (`'all' | 'onlyCurrentTree'| 'onlySiblings'` , por defecto: `'all'`):
 - `'all'` : Muestra todos los items.
 - `'onlyCurrentTree'` : Muestra solo los items que están en el árbol actual (el ítem activo, su padre y sus hijos)
 - `'onlySiblings'` : Muestra solo los items que están en el árbol actual y sus hijos directos.

Link

Inserta un enlace que pueda utilizar para navegar a una diapositiva determinada.

Uso

```
<Link to="42">Ir a la diapositiva 42</Link>  
<Link to="42" title="Ir a la diapositiva 42"/>
```

Parámetros:

- `to (string | number)`: La ruta de la diapositiva a la que se va a navegar (las diapositivas empiezan por `1`)
- `title (string)`: El título a mostrar

Titles

Inserta el título principal de una diapositiva analizada como HTML.

Los títulos y los niveles de título se heredan automáticamente del primer elemento de título de cada diapositiva.

Puedes sobreescibir este comportamiento en cada diapositiva usando la sintaxis de frontmatter:

```
---
```

`title: Mi presentación`
`level: 2`

```
---
```

Uso

El componente `<Titles>` es un componente virtual que se puede importar con:

```
import Titles from '@/slidev/titles.md'
```

Y puedes usarlo así:

```
<Titles no="42" />
```

Parámetros

- `no (string | number)`: El numero de la diapositiva de la cual extraer el título (las diapositivas empiezan por `1`)

Componentes personalizados

Crea un directorio llamado `components/` en la raíz de tu proyecto, y simplemente mueve allí los componentes de Vue. Una vez lo hagas, podrás usarlo con el mismo nombre que en tu archivo de Markdown.

Lea más en la sección de [personalización](#)

Componentes proveídos por temas

Los temas pueden también proveer sus propios componentes. Por favor, lea su documentación para aquello que provean.

Lea más en la sección de [estructura del contenido](#).

[Go to TOC](#)

Plantillas

Plantillas integradas

As themes may override layouts behaviour, the best way to know exactly the usage, parameters and examples is referring their documentation.

center

Displays the content in the middle of the screen.

cover

Used to display the cover page for the presentation, may contain the presentation title, contextualization, etc.

default

The most basic layout, to display any kind of content.

end

The final page for the presentation.

fact

To show some fact or data with a lot of prominence on the screen.

full

Use all the space of the screen to display the content.

image-left

Shows an image on the left side of the screen, the content will be placed on the right side.

Usage

```
---
```

```
layout: image-left
```

```
# the image source
```

```
image: ./path/to/the/image
```

```
# a custom class name to the content
```

```
class: my-cool-content-on-the-right
```

```
--
```

image-right

Shows an image on the right side of the screen, the content will be placed on the left side.

Usage

```
---  
layout: image-right  
  
# the image source  
image: ./path/to/the/image  
  
# a custom class name to the content  
class: my-cool-content-on-the-left  
---
```

image

Shows an image as the main content of the page.

Usage

```
---  
layout: image  
  
# the image source  
image: ./path/to/the/image  
---
```

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

To introduce the presentation, usually with the presentation title, a short description, the author, etc.

none

A layout without any existent styling.

quote

To display a quotation with prominence.

section

Used to mark the beginning of a new presentation section.

statement

Make an affirmation/statement as the main page content.

two-cols

Separates the page content in two columns.

Usage

```
---  
layout: two-cols  
---  
  
# Left  
  
This shows on the left  
  
::right::
```

Right

This shows on the right

Plantillas personalizadas

Crea un directorio `layouts/` en la raíz de tu proyecto, y simplemente pon los componentes de la plantilla (layout) Vue en él.

Lee más en la sección de [personalización](#)

Plantillas proveídas por temas

Los temas pueden proveer plantillas o reemplazar aquellas existentes. Por favor, lea su documentación para aquello que provean.

[Go to TOC](#)

Configurar KaTeX

Crea `./setup/katex.ts` con el siguiente contenido:

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

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

Con la configuración, puedes proporcionar la configuración personalizada para las [opciones de KaTeX](#). Consulta las definiciones de tipo y su documentación para obtener más detalles.

Configurar Mermaid

Crea `./setup/mermaid.ts` con el siguiente contenido:

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

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

Con la configuración, puede proporcionar una configuración personalizada por defecto para [Mermaid](#). Consulta las definiciones de tipo y su documentación para obtener más detalles.

Configurar Monaco

Crea `./setup/monaco.ts` con el siguiente contenido:

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

export default defineMonacoSetup(async (monaco) => {
  // usa `monaco` para configurarlo
})
```

Revisa más información sobre la [configuración de Mónaco](#).

Uso

Para usar Monaco en tus diapositivas, simplemente añade `{monaco}` a tus fragmentos de código:

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

console.log(plusOne.value) // 2

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

A

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

console.log(plusOne.value) // 2

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

Exportando

Por defecto, Monaco sólo funcionará en modo `dev`. Si también quieres tenerlo disponible en el SPA exportado, puedes configurarlo en tu frontmatter:

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

Instalación automática de tipos

Cuando se utiliza TypeScript con Monaco, los tipos para las dependencias se instalarán en el lado del cliente automáticamente.

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

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

En el ejemplo anterior, solo asegúrate de que `vue` y `@vueuse/core` están instalados localmente bajo `dependencies` / `devDependencies`, iSlidev se encargará del resto y tu editor funcionará mágicamente!

Configurar un tema

El tema es controlado por Slidev basado en el tema claro/oscuro. Si quieres personalizarlo, puedes pasar el id del tema a las funciones de configuración:

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

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

Si quieres cargar temas personalizados:

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

// cambios a tus temas
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',
    },
  }
})
```

Si estás creando un tema para Slidev, puedes usar el `import()` dinámico dentro de la función de configuración para obtener resultados con tree-shaking o code-splitting.

Configurar atajos

Disponible desde v0.20

Crea `./setup/shortcuts.ts` con el siguiente contenido:

```
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,
    },
  ],
})
```

Con esta configuración, puedes dar ajustes personalizados a los atajos mencionados en [Navegación](#). La configuración anterior vincula la siguiente animación o diapositiva a `enter` y la animación o diapositiva anterior a `retroceso`.

La función de configuración recibe un objeto con algunos métodos de navegación, y devuelve un array que contiene alguna configuración de accesos directos. Consulta las definiciones tipo para más detalles.

Echa un vistazo a [useMagicKeys | VueUse](#) para más detalles sobre el evento de tecla pulsada.

Configurar Vite

Slidev es impulsado por [Vite](#) detrás de las escenas. Esto significa que puedes aprovechar el gran sistema de plugins de Vite para personalizar aún más tus diapositivas.

`vite.config.ts` será respetado si tienes uno.

Slidev tiene los siguientes plugins preconfigurados:

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

Más información sobre las [preconfiguraciones aquí](#).

Configurar plug-ins internos

Disponible desde v0.21

Para configurar los plug-ins internos listados más arriba, crea `vite.config.ts` con el siguiente contenido. Por favor, ten en cuenta que Slidev tiene algunas opciones de preconfiguración para esos plugins, haciendo esto podría desconfigurar alguno de ellos, lo que podría causar que la aplicación se rompa. Por favor, trata esto como **una característica avanzada**, asegúrate de saber lo que estás haciendo antes de seguir adelante.

```
import { defineConfig } from 'vite'

export default defineConfig({
  slidev: {
    vue: {
      /* opciones de vue */
    },
    markdown: {
      /* opciones de markdown-it */
      markdownItSetup(md) {
        /* plugins de markdown-it personalizados */
        md.use(/* ... */)
      },
      /* opciones para otros plugins */
    },
  },
})
```

Echa un vistazo a las [declaraciones de tipos](#) para más opciones.

[Go to TOC](#)

Configurar Vue

Slidev utiliza [Vue 3](#) para renderizar la aplicación en el lado del cliente. Puedes extender la aplicación para añadir plugins o configuraciones personalizadas.

Crea `./setup/main.ts` con el siguiente contenido:

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

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

Esto también podría ser utilizado como la entrada principal de tu aplicación Slidev para hacer algunas inicializaciones antes de que la aplicación se inicie.

Más información: [Vue Application API](#).

Configurar Windi CSS

Markdown soporta por defecto las etiquetas HTML incrustadas. Por lo tanto, puedes estilizar tu contenido de la manera que quieras. Por conveniencia, tenemos [Windi CSS](#) incorporado, por lo que puedes estilizar el marcado directamente usando clases de utilidades.

Por ejemplo:

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

El modo de [attributify](#) en [Windi CSS v3.0](#) está activado por defecto.

Configuraciones

Para configurar Windi CSS, crea `setup/windicss.ts` con el siguiente contenido para extender las configuraciones por defecto:

```
// setup/windicss.ts

import { defineWindiSetup } from '@slidev/types'

// extendiendo las configuraciones por defecto de windicss
export default defineWindiSetup(() => ({
  shortcuts: {
    // personalizar el fondo por defecto
    'bg-main': 'bg-white text-[#181818] dark:(bg-[#121212] text-[#ddd])',
  },
  theme: {
    extend: {
      // las fuentes pueden ser reemplazadas aquí, recuerda actualizar los enlaces
      // de las fuentes web en `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',
      },
    },
  },
}))
```

Lee más sobre [las configuraciones de Windi CSS](#).

[Go to TOC](#)

Estructura de directorios

Slidev emplea algunas convenciones de estructura de directorios para minimizar la superficie de configuración y hacer que las extensiones de funcionalidad sean flexibles e intuitivas.

La estructura básica es la siguiente:

```
tu-slidev/
  |-- components/      # componentes personalizados
  |-- layouts/         # plantillas personalizadas
  |-- public/          # archivos estáticos
  |-- setup/           # configuración personalizada / hook
  |-- styles/          # estilo personalizado
  |-- index.html        # inyecciones a index.html
  |-- slides.md         # la entrada principal de las diapositivas
  |-- vite.config.ts    # extender la configuración de vite```

```

Todas ellas son opcionales.

Componentes

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

Los componentes dentro de este directorio pueden ser utilizados directamente en las diapositivas Markdown con el mismo nombre de componente que el nombre del archivo.

Por ejemplo:

```
```bash
tu-slidev/
 |-- ...
 |-- components/
 |-- MyComponent.vue
 |-- HelloWorld.ts
```

```

```
<!-- slides.md -->
# Mi presentación
<MyComponent :count="4"/>
<!-- ambas formas funcionan -->
<hello-world foo="bar">
  Slot
</hello-world>
```

Esta función usa `vite-plugin-components`, aprende más ahí.

Slidev también provee [componentes por defecto](#) para tu uso.

Plantillas

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

```
tu-slidev/
  ...
  └── layouts/
    ├── cover.vue
    └── my-cool-theme.vue
```

Puedes usar cualquier nombre de archivo para tu diseño. A continuación, haz referencia a tu diseño en tu cabecera YAML utilizando el nombre del archivo.

```
---
layout: my-cool-theme
---
```

Si el diseño que proporcionas tiene el mismo nombre que una plantilla por defecto o una plantilla de un tema, tu diseño personalizado tendrá prioridad sobre ellos. El orden de prioridad es `local > tema > integrado`.

En la plantilla, utilice `<slot/>` para el contenido de la diapositiva. Por ejemplo:

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

Público

Convenciones: `./public/*`

Los activos de este directorio se servirán en la ruta raíz `/` durante el desarrollo, y se copiarán en la raíz del directorio dist tal cual. Más información sobre [el directorio public de Vite](#).

Estilo

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

Los archivos que sigan esta convención se injectarán en la raíz de la aplicación. Si necesitas importar varias entradas css, puedes crear la siguiente estructura y gestionar tú mismo el orden de importación.

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

```
// styles/index.ts
import './base.css'
import './code.css'
import './layouts.css'
```

Los estilos serán procesados por [Windi CSS](#) y [PostCSS](#), así que puedes usar CSS anidado y [directivas @](#) por defecto. Por ejemplo:

```
.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');
  }
}
```

Lee más sobre la [sintaxis](#).

index.html

Convenciones: `index.html`

El `index.html` ofrece la posibilidad de inyectar etiquetas meta y/o scripts al `index.html` principal

Por ejemplo, para el siguiente `index.html` personalizado:

```
<!-- ./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="./your-scripts"></script>
</body>
```

El `index.html` final 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=".//your-scripts"></script>
</body>
</html>
```

Global Layers

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

Learn more: [Global Layers](#)

[Go to TOC](#)

Fuentes

Disponible desde v0.20

Aunque puedes utilizar HTML y CSS para personalizar las fuentes y el estilo de tus diapositivas como quieras, Slidev también te ofrece una forma cómoda de utilizarlas sin esfuerzo alguno.

En tu *frontmatter*, configúralo de la siguiente manera:

```
---  
  fonts:  
    # normalmente, el texto  
    sans: 'Robot'  
    # utilizado junto a la clase 'font-serif' de windicss  
    serif: 'Robot Slab'  
    # para bloques de código, código en línea, etc.  
    mono: 'Fira Code'  
---
```

Y eso es todo.

Las fuentes serán **importadas automáticamente desde Google Fonts**. Eso significa que puedes utilizar cualquier fuente alojada en Google Fonts.

Fuentes locales

Por defecto, Slidev asume que todas las fuentes especificadas a través de las configuraciones `fonts` provienen de Google Fonts. Si deseas utilizar fuentes locales, especifica `fonts.local` para evitar que colisione con Google Fonts.

```
---  
  fonts:  
    # Igual que font-family en CSS, puedes separar con una coma para el fallback  
    sans: 'Helvetica Neue, Robot'  
    # marca 'Helvetica Neue' como una fuente local  
    local: 'Helvetica Neue'  
---
```

Grosores e itálicas

Por defecto, Slidev viene con tres grosores: `200, 400, 600` para cada fuente. Puedes configurarlo mediante:

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

```
---  
# importa fuentes itálicas, `false` por defecto  
italic: false
```

Esta configuración se aplica a todas las fuentes web. Para un control más preciso de los grosores de cada fuente, deberás importarlos manualmente con [HTML](#) y CSS.

Fuentes fallback

En la mayoría de los casos, bastará con marcar la "fuente especial" y Slidev se encargará de añadir el resto de fallbacks. Por ejemplo:

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

resultará en:

```
.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;  
}
```

Si quieres desactivar este comportamiento, puedes hacerlo con:

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

Proveedores

- Opciones: `google` | `none`
- Por defecto: `google`

Por el momento, solo se soporta Google como proveedor de fuentes aunque está planeado integrar otros en el futuro. Específicalo como `none` para desactivar toda la funcionalidad de auto-importes y depender de fuentes locales.

```
---  
  fonts:  
    provider: 'none'  
---
```

Capas globales

Available since v0.17

Las capas globales le permiten tener componentes personalizados que **persisten** a través de las diapositivas. Esto podría ser útil para tener pies de página, animaciones entre diapositivas, efectos globales, etc.

Slidev proporciona tres capas para este uso, crea `global-top.vue` o `global-bottom.vue` o `custom-nav-controls.vue` en la raíz de tu proyecto y lo detectará automáticamente.

Relación de capas:

- Global superior (`global-top.vue`)
- Diapositivas
- Global inferior (`global-bottom.vue`)
- Controles de navegación
 - Controles de navegación personalizados (`custom-nav-controls.vue`)

Ejemplo

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

El texto `Tu nombre` aparecerá en todas tus diapositivas.

```
<!-- custom-nav-controls -->
<template>
  <button class="icon-btn" title="Siguiente" @click="$slidev.nav.next">
    <carbon:arrow-right />
  </button>
</template>
```

El botón "Siguiente" aparecerá en la navegación.

Para habilitarlo condicionalmente, puedes aplicarlo con el [contexto global de Vue](#).

```
<!-- oculta el pie de página de la página 4 -->
<template>
  <footer
    v-if="$slidev.nav.currentPage !== 4"
    class="absolute bottom-0 left-0 right-0 p-2">
    >
    Tu nombre
  </footer>
</template>
```

```
<!-- ocultar el pie de página de la plantilla "portada" -->
<template>
  <footer>
    v-if="$slidev.nav.currentLayout !== 'cover'"
    class="absolute bottom-0 left-0 right-0 p-2"
  >
    Tu nombre
  </footer>
</template>
```

```
<!-- un ejemplo de pie de página -->
<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>
```

```
<!-- custom-nav-controls -->
<!-- oculta el botón en el modo presentador -->
<template>
  <button v-if="!$slidev.nav.isPresenter" class="icon-btn" title="Next"
@click="$slidev.nav.next">
  <carbon:arrow-right />
</button>
</template>
```

Resaltadores

Slidev viene con dos resaltadores de sintaxis para que tú puedes elegir:

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

Prism es uno de los resaltadores de sintaxis más populares. El resaltado se hace añadiendo clases de token al código y se colorea usando CSS. Puedes navegar a través de sus [temas oficiales](#), o crear/personalizar uno tú mismo muy fácilmente usando `prism-theme-vars`.

Shiki, por otro lado, es un resaltador de sintaxis basado en la gramática de TextMate. Genera tokens de colores, por lo que no se necesita CSS adicional. Dado que tiene un gran soporte gramatical, los colores generados son muy precisos, igual que lo que verás en VS Code. Shiki también viene con [un montón de temas incorporados](#). La desventaja de Shiki es que también requiere temas de TextMate (compatibles con el tema de VS Code) para hacer el resaltado, lo que puede ser un poco más difícil de personalizar.

Los temas de Slidev normalmente soportan tanto Prism como Shiki, pero dependiendo del tema que estés usando, puede que sólo soporte uno de ellos.

Cuando tienes la opción, el intercambio es básicamente:

- **Prism** para una personalización más fácil
- **Shiki** para un resaltado más preciso

Por defecto, Slidev utiliza Prism. Puedes cambiarlo modificando tu frontmatter:

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

Configurar Prism

Para configurar tu Prism, puedes simplemente importar el tema css o usar `prism-theme-vars` para configurar los temas para el modo claro y oscuro. Consulta su documentación para más detalles.

Configurar Shiki

Create `./setup/shiki.ts` file with the following content

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

```
  },
})
```

Consulta los nombres de los temas disponibles en [Shiki's docs](#).

O si quieres usar tu propio tema:

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

import { defineShikiSetup } from '@slidev/types'

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

Personalizaciones

Slidev es totalmente personalizable, desde el estilo hasta la configuración de las herramientas. Permite configurar las herramientas que funcionan "detrás de las escenas" ([Vite](#), [Windi CSS](#), [Monaco](#), etc.)

Configuración de Frontmatter

Puedes configurar Slidev en el frontmatter de tu primera diapositiva, a continuación se muestra el valor por defecto de cada opción.

```
---
# ID del tema o nombre del paquete
theme: 'default'
# título de tu diapositiva, se deducirá automáticamente de la primera cabecera si no se especifica
title: 'Slidev'
# titleTemplate for the webpage, `'%s` will be replaced by the page's title
titleTemplate: '%s - Slidev'
# información para tus slides, puede ser un string markdown
info: false

# descarga habilitada de pdf en la compilación de SPA, también puede ser una url personalizada
download: false
# nombre del archivo exportado
exportFilename: 'slidev-exported.pdf'
# resaltador de sintaxis, puede ser 'prisma' o 'shiki'
highlighter: 'prism'
# muestra el número de línea
lineNumbers: false
# activa el editor monaco, puede ser booleano, 'dev' o 'build'
monaco: 'dev'
# download remote assets in local using vite-plugin-remote-assets, can be boolean, 'dev' or 'build'
remoteAssets: false
# controla si los textos en las diapositivas son seleccionables
selectable: true
# habilita grabación de diapositivas, puede ser booleano, 'dev' o 'build'
record: 'dev'

# forzar el esquema de colores para las diapositivas, podría ser 'auto', 'light', o 'dark'
colorSchema: 'auto'
# modo del router para vue-router, podría ser "history" o "hash"
routerMode: 'history'
# relación de aspecto de las diapositivas
aspectRatio: '16/9'
# ancho real del lienzo, unidad en px
canvasWidth: 980
# usado para la personalización del tema, injectará estilos raíz como `--slidev-theme-x` para la atribución `x`
themeConfig:
  primary: '#5d8392'

# favicon, puede ser una ruta local o una URL
favicon: 'https://cdn.jsdelivr.net/gh/slidevjs/slidev/assets/favicon.png'
```

```

# URL del servidor de PlantUML para renderizar diagramas
plantUmlServer: 'https://www.plantuml.com/plantuml'
# fuentes que se importarán desde Google Fonts
# Leer más: https://es.sli.dev/custom/fonts
fonts:
  sans: 'Roboto'
  serif: 'Roboto Slab'
  mono: 'Fira Code'

# frontmatter aplicado por defecto a todas las diapositivas
defaults:
  layout: 'default'
  # ...

# opciones de dibujado
# Para leer más: https://es.sli.dev/guide/drawing.html
drawings:
  enabled: true
  persist: false
  presenterOnly: false
  syncAll: true
---
```

Revisa las [definiciones de tipos](#) para más opciones.

Estructura del directorio

Slidev utiliza convenciones de estructura de directorios para minimizar la superficie de configuración y hacer que las extensiones en la funcionalidad sean flexibles e intuitivas.

Consulta la sección de [estructura de directorios](#).

Herramientas de configuración

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

[Go to TOC](#)

Contexto global de Vue

Slidev inyecta un [contexto global de Vue](#) `$slidev` para condiciones avanzadas o controles de navegación.

Uso

Puedes acceder a ello en cualquier parte de tu Vue o Markdown, con [la sintaxis "Mustache"](#).

```
<!-- slides.md -->
# Página 1
La página actual es: {{ $slidev.nav.currentPage }}
```

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

Propiedades

`$slidev.nav`

Un objeto reactivo que contiene las propiedades y controles de la navegación de las diapositivas. Por ejemplo:

```
$slidev.nav.next() // ir al siguiente paso
$slidev.nav.nextSlide() // ir a la siguiente diapositiva (saltar v-clicks)
$slidev.nav.go(10) // ir a la diapositiva #10
```

```
$slidev.nav.currentPage // número de la diapositiva actual
$slidev.nav.currentLayout // plantilla (layout) actual
$slidev.nav.clicks // contador actual de clics
```

Para conocer más propiedades disponibles, consulta los miembros exportados por [nav.ts](#).

`$slidev.configs`

Un objeto reactivo que contiene las [configuraciones en el primer frontmatter](#) analizadas de tu `slides.md`. Por ejemplo

```
---  
title: ¡Mi primer Slidev!
```

```
{{ $slidev.configs.title }} // '¡Mi primer Slidev!'
```

\$slidev.themeConfigs

Un objeto reactivo que contiene las configuraciones analizadas del tema.

```
---  
title: ¡Mi primer Slidev!  
themeConfig:  
  primary: #213435  
---
```

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

Animaciones

Animaciones en clic

v-click

Para aplicar "animaciones en clic" a los elementos, puedes usar la directiva `v-click` o los componentes `<v-click>`

```
# Hola
<!-- Uso del componente: esto estará invisible hasta que presiones "siguiente" -->
<v-click>

Hola Mundo
</v-click>

<!-- Uso de la directiva: esto estará invisible hasta que presiones "siguiente"
por segunda vez -->
<div v-click class="text-xl p-2">
    ¡Hey!
</div>
```

v-after

El uso de `v-after` es similar al de `v-click` pero hará visible el elemento cuando se active el `v-click` anterior.

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

Al hacer clic en el botón "siguiente", tanto `Hola` como `Mundo` aparecerán juntos.

v-click-hide

Same as `v-click` but instead of making the element appear, it makes the element invisible after clicking.

```
<div v-click-hide>Hello</div>
```

v-clicks

`v-clicks` sólo se proporciona como un componente. Es una forma abreviada de aplicar la directiva `v-click` a todos sus elementos descendientes. Es especialmente útil cuando se trabaja con listas.

```
<v-clicks>
    - Ítem 1
    - Ítem 2
    - Ítem 3
```

- Ítem 4

```
</v-clicks>
```

Cada vez que haga clic en "siguiente" se hará visible un elemento.

Recuento de clics personalizado

Por defecto, Slidev cuenta cuántos pasos son necesarios antes de pasar a la siguiente diapositiva. Puedes anular esta configuración pasando la opción de en Front Matter `clicks`:

```
---  
# 10 clics en esta diapositiva, antes de pasar a la siguiente  
clicks: 10  
---
```

Ordenando

Pasando el índice de `v-click` a tus directivas, puedes personalizar el orden de mostrado

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

```
<!-- el orden se invierte -->  
<div v-click="3">1</div>  
<div v-click="2">2</div>  
<div v-click="1">3</div>
```

```
---  
clicks: 3  
---  
  
<!-- visible después de 3 clics -->  
<v-clicks at="3">  
  <div>Hola</div>  
</v-clicks>
```

Transiciones de elementos

Cuando aplique la directiva `v-click` a sus elementos, se le adjuntará el nombre de clase `slidev-vclick-target`. Cuando los elementos están ocultos, el nombre de la clase `slidev-vclick-hidden` también se adjuntará. Por ejemplo:

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

Después de un clic, se convertirá en:

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

Por defecto, se aplica una sutil transición de opacidad a esas clases:

```
// the default

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

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

Puedes sobreescribirlos para personalizar los efectos de transición en tus hojas de estilo personalizadas.

Por ejemplo, puedes conseguir las transiciones de escalado mediante:

```
// styles.css

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

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

Para especificar animaciones sólo para determinadas diapositivas o diseños:

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

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

Aprende más sobre [personalizar estilos](#)

Movimiento

Slidev tiene [@vueuse/motion](#) incorporado. Puedes usar la directiva `v-motion` a cualquier elemento para hacer que se le aplique movimiento. Por ejemplo:

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

El texto `Slidev` se moverá de `-80px` a su posición original en la inicialización.

Nota: Sliddev precarga la siguiente diapositiva para mejorar el rendimiento, lo que significa que las animaciones podrían comenzar antes de que usted navegue por la página. Para que funcione correctamente, puede desactivar la precarga para la diapositiva en particular

```
---  
preload: false  
---
```

O controla el ciclo de vida de los elementos con `v-if` para tener controles precisos

```
<div  
  v-if="$slidev.nav.currentPage === 7"  
  v-motion  
  :initial="{ x: -80 }"  
  :enter="{ x: 0 }">  
  Sliddev  
</div>
```

Aprende más: [Demo](#) | [@vueuse/motion](#) | [v-motion](#) | [Presets](#)

Transiciones de páginas

La versión actual NO incluye soporte para diapositivas. Estamos planeando añadir soporte para ellas en la próxima versión mayor. Mientras tanto, puedes utilizar tus estilos y librerías personalizadas para hacerlo.

[Go to TOC](#)

Dibujos y anotaciones

Disponible desde v0.23

We have [drauu](#) built-in for drawing and annotation that could enhance your presentation further.

To start, click the icon in the toolbar and start drawing. It's also available in the [Presenter Mode](#). Drawings and annotations you created will be **synced up** automatically across all instances in real-time.

Use with Stylus Pen

When using a stylus pen on a tablet (for example, iPad with Apple Pencil), Slidev could smartly detect the input type. You can directly draw on your slides with the pen without turning on the drawing mode, while having your fingers or mouse control the navigation.

Persist Drawings

The following frontmatter configuration allows you to persist your drawings as SVGs under `.slidev/drawings` directory and have them inside your exported pdf or hosted site.

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

Disable Drawings

Entirely:

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

Only in Development:

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

Only in Presenter Mode:

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

Drawing Syncing

By default, Slidev syncs up your drawings across all instances. If you are sharing your slides with others, you might want to disable the syncing by:

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

With this config, only the drawing from the presenter instance will be able to sync with others.

Soporte de editores

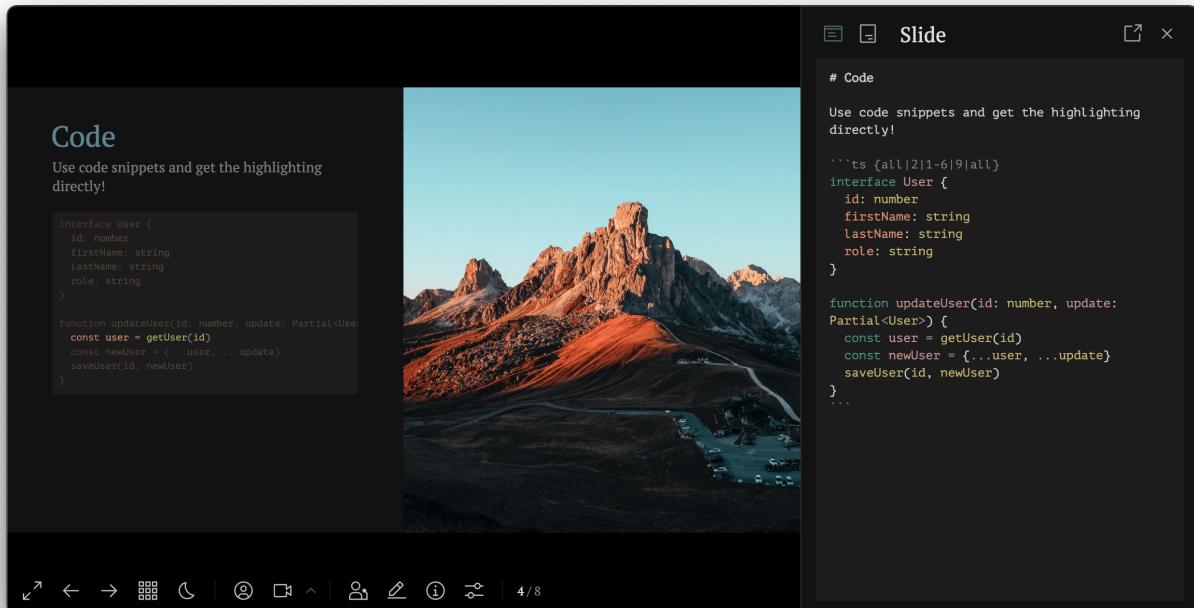
Dado que Slidev utiliza Markdown como entrada de origen, puedes utilizar CUALQUIER editor que te guste para escribirlo.

Si deseas un manejo de alto nivel a tus diapositivas, ¡hemos proporcionado las siguientes integraciones de editores!

Editor integrado

Slidev viene con un editor integrado [CodeMirror](#) que instantáneamente recargará y guardará los cambios en tu archivo.

Haz clic en el botón para abrirlo.



Extensión de VS Code



Slidev for VS Code

 VS Code Marketplace v0.4.1

downloads 26k

La extensión de VS Code ofrece algunas funciones que te ayudarán a organizar mejor tus diapositivas y a tener una visión general rápida de las mismas.

Características

- Ver diapositivas en el panel lateral
- Botones para ir a la siguiente/anterior
- Reordenación de diapositivas
- Plegado de diapositivas
- Convertir Markdown a HTML

The screenshot shows a code editor window titled "slides.md — demo". The left sidebar displays a table of contents (TOC) under the heading "SLIDEV: SLIDES". The TOC includes sections like Composable Vue, Anthony Fu, Sponsors, Composable Vue, VueUse, Composition API, Ref, Ref Auto Unwrapping, 'unref' - Oppsite of Ref, Patterns & Tips, What's Composable Func..., Think as "Connections", One Thing at a Time, Passing Refs as Argumen..., MaybeRef <MarkerTips/>, Make it Flexible <MarkerP..., 'useTitle' <Marker class=..., "Reuse" Ref <MarkerCore..., 'ref' / 'unref' <MarkerTi..., Object of Refs <MarkerPa..., Async to "Sync" <Marker..., 'useFetch' <Marker clas..., and Side-effects Self Cleanup....

The main editor area contains the following code:

```
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">
```

Markdown

---

[Go to TOC](#)

# Exportando

## PDF

La exportación a PDF o PNG se basa en [Playwright](#) para la renderización. Por lo tanto, tendrás que instalar `playwright-chromium` para utilizar esta función. Si estás haciendo la exportación en un entorno CI, [la guía CI de playwright](#) puede ser útil.

Instala `playwright-chromium`

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

Ahora exporta sus diapositivas a PDF utilizando el siguiente comando:

```
$ slidev export
```

Después de unos segundos, tus diapositivas estarán listas en `./slides-export.pdf`.

En el caso que quieras exportar tus diapositivas usando la versión oscura del tema, utiliza la opción `--dark`:

```
$ slidev export --dark
```

## Exportar pasos por clic

Disponible desde v0.21

Por defecto, Slidev exporta 1 página por diapositiva con las animaciones de los clics desactivadas. Si deseas exportar diapositivas con múltiples pasos en múltiples páginas, agrega `--with-clicks`.

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

## PNGs

Al pasar la opción `--format png`, Slidev exportará imágenes PNG para cada diapositiva en lugar de un PDF.

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

## Single-Page Application (SPA)

Revisa [Alojamiento Estático](#).

---

[Go to TOC](#)

# FAQ

## Grids

Dado que Slidev está basado en la web, puedes aplicar cualquier modelo de grid como [CSS Grids](#), [flexboxes](#), o incluso [Masonry](#), tú tienes el control.

Dado que tenemos [Windi CSS](#) integrado, aquí te mostramos una forma simple como referencia:

```
<div class="grid grid-cols-2 gap-4">
<div>
 La primera columna
</div>
<div>
 La segunda columna
</div>
</div>
```

Yendo más allá, puedes personalizar el tamaño de cada columna con:

```
<div class="grid grid-cols-[200px,1fr,10%] gap-4">
<div>
 La primera columna (200px)
</div>
<div>
 La segunda columna (auto ajuste)
</div>
<div>
 La tercera columna (10% del ancho del elemento padre)
</div>
</div>
```

Lee más sobre [Windi CSS Grids](#).

## Posicionamiento

Las diapositivas se definen en tamaños fijos (por defecto [980x552px](#)) y se ajustan a la pantalla del usuario. Puedes utilizar con seguridad la posición absoluta en tus diapositivas ya que se escalarán junto con la pantalla.

Por ejemplo:

```
<div class="absolute left-30px bottom-30px">
 This is a left-bottom aligned footer
</div>
```

Para cambiar el tamaño real del lienzo, puedes pasar las opciones `canvasWidth` en tu primer frontmatter:

```

 canvasWidth: 800

```

## Tamaño de la letra

Si crees que el tamaño de la fuente en tus diapositivas es demasiado pequeño, puedes ajustarlo de varias maneras:

### Anular el estilo local

Puedes anular los estilos de cada diapositiva con la etiqueta `<style>`.

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

Página 2
No se verá afectado
```

Leer más: [Estilos Incrustados](#)

### Reemplazando estilos globales

Puedes crear tus propios estilos globales en `./style.css`, por ejemplo

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

Leer más: [Global Style](#)

### Escalar el lienzo

Al cambiar el tamaño real del lienzo se escalarán todos los contenidos (texto, imágenes, componentes, etc.) y las diapositivas

```

por defecto: 980
como el lienzo se hace más pequeño, el tamaño visual será mayor
canvasWidth: 800

```

## Usar Transform

Disponemos de una funcionalidad integrada llamada `<Transform />`, que envuelve la propiedad transform de CSS.

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

# Alojamiento estático

## Construye Single Page Applications (SPA)

Puedes convertir las diapositivas en una SPA:

```
$ slidev build
```

La app generada se encontrará en `dist/`, que podrás alojarla en [GitHub Pages](#), [Netlify](#), [Vercel](#), o donde quieras. Ahora puedes presentar tus diapositivas con el resto del mundo con tan solo un link.

### Ruta base

Para desplegar tus diapositivas bajo sub-rutas, necesitarás usar la opción `--base`. Por ejemplo:

```
$ slidev build --base /charlas/mi-super-charla/
```

Revisa la [documentación de Vite](#) para más detalles.

### PDF descargable

Puedes generar un PDF descargable para tus usuarios con esta configuración:

```

 download: true

```

Slidev generará un archivo PDF junto al resto de archivos, y un botón de descargar se mostrará en la SPA. También puedes dar un link personalizado, en tal caso se omitirá la generación del PDF.

```

 download: 'https://mipagina.com/mi-charla.pdf'

```

## Ejemplos

Aquí hay algunos ejemplos de SPAs exportadas:

- [Plantilla de Inicio](#)
- [Composable Vue por Anthony Fu](#)

Si quieres ver más, echa un vistazo a la [galería](#).

## Alojamiento

Recomendamos usar `npm init slidev@lastest` para generar tu proyecto, que contiene los archivos de configuración necesarios para servicios de alojamiento más populares.

## Netlify

- [Netlify](#)

Crea `netlify.toml` en tu carpeta con el siguiente contenido:

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

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

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

Después, en tu panel de Netlify crea un nuevo sitio con esos archivos.

## Vercel

- [Vercel](#)

Crea `vercel.json` en tu proyecto con la siguiente configuración.

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

Después, en tu panel de Vercel crea un nuevo sitio con esos archivos.

## GitHub Pages

- [GitHub Pages](#)

Crea `.github/workflows/deploy.yml` con el siguiente contenido para automatizar el despliegue de diapositivas con Github Pages mediante Github Actions.

```
name: Desplegar páginas
on: push
jobs:
 deploy:
 runs-on: ubuntu-latest
 steps:
 - uses: actions/checkout@v2
 - uses: actions/setup-node@v2
 with:
 node-version: '14'
 - name: Instalando dependencias
 run: npm install
 - name: Construyendo
 run: npm run build
 - name: Desplegando
```

```
uses: crazy-max/ghaction-github-pages@v2
with:
 build_dir: dist
env:
 GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
```

# Primeros pasos

## Introducción

Slidev (slide + dev, `/slайдив/`) es un creador y presentador de diapositivas basado en la web. Está diseñado para que los desarrolladores se centren en la escritura de contenido en Markdown, mientras que también tiene el poder de los componentes HTML y Vue para ofrecer diseños y plantillas *pixel-perfect* con demostraciones interactivas incrustadas en sus presentaciones.

Utiliza un archivo Markdown rico en funciones para generar diapositivas bonitas con una experiencia de recarga instantánea, junto con muchas integraciones incorporadas como la escritura de código en vivo, la exportación de PDF, la grabación de presentaciones, etc. Dado que está impulsado por la web, puedes hacer cualquier cosa con Slidev: las posibilidades son infinitas.

Puedes obtener más información sobre la justificación del proyecto en la sección de [Por qué Slidev](#).

## Características

- 📝 **Basado en Markdown** - utiliza tus editores y forma de trabajar favoritos.
- 💻 **Pensado para desarrolladores** - resaltado de sintaxis integrado, escritura de código en vivo, etc.
- 🎨 **Tematizable** - un tema puede ser compartido y usado con paquetes npm.
- 🌈 **Con estilo** - Utilidades de [Windi CSS](#) bajo demanda, hojas de estilos fáciles de incrustar.
- 🎭 **Interactivo** - incrusta componentes de Vue sin problemas
- 📱 **Modo Presentador** - usa otra ventana, o incluso tu teléfono para controlar las diapositivas
- 🖌️ **Drawing** - dibuja y anota en tus diapositivas
- TeX **LaTeX** - soporte integrado para ecuaciones matemáticas en LaTeX
- Diagrama **Diagramas** - crea diagramas con descripciones textuales
- 💡 **Iconos** - acceso directo a iconos de cualquier set
- 💻 **Editores** - editor integrado, o [extensión para VS Code](#)
- 🎥 **Grabar** - grabación y vista de cámara integrados
- 📦 **Portable** - exporta a PDF, PNGs, o incluso un SPA alojable
- ⚡ **Rápido** - recarga instantánea impulsada por [Vite](#)
- 🛠️ **Hackable** - usando plugins de Vite, componentes de Vue, o cualquier paquete de npm

## Tecnologías

Slidev es posible gracias a la combinación de estas herramientas y tecnologías.

- [Vite](#) - Una herramienta de frontend extremadamente rápida
- [Vue 3](#) potenciado por [Markdown](#) - Céntrate en el contenido mientras dispones de la potencia de los componentes HTML y Vue siempre que lo necesites
- [Windi CSS](#) - Framework de CSS de utilidades bajo demanda, estilizando tus diapositivas con facilidad
- [Prism](#), [Shiki](#), [Monaco Editor](#) - Fragmentos de código con edición en vivo de primera clase
- [RecordRTC](#) - Grabación y vista de cámara integrados
- [VueUse](#) - `@vueuse/core`, `@vueuse/head`, `@vueuse/motion`, etc.

- [Iconify](#) - Colecciones de iconos.
- [Draau](#) - Soporte de dibujos y anotaciones.
- [KaTeX](#) - Renderizado matemático de LaTeX.
- [Mermaid](#) - Diagramas textuales.

## Preparando tu primera presentación

### Pruébalo online

[sli.dev/new](#)



[Open in StackBlitz](#)

### Créalo localmente

Con NPM:

```
$ npm init slidev
```

Con Yarn:

```
$ yarn create slidev
```

Sigue las instrucciones, iy empieza a crear tus diapositivas ahora! Para obtener más detalles sobre la sintaxis de Markdown, lee la [guía de sintaxis](#).

## Interfaz de la línea de comandos

En un proyecto donde Slidev está instalado, puedes usar el binario `slidev` en tus scripts npm.

```
{
 "scripts": {
 "dev": "slidev", // start dev server
 "build": "slidev build", // build for production SPA
 "export": "slidev export" // export slides to pdf
 }
}
```

Sino, puedes usarlo con `npx`

```
$ npx slidev
```

Ejecuta `slidev --help` para ver más opciones disponibles.

## Sintaxis de Markdown

Slidev lee tu archivo `slides.md` bajo la raíz de tu proyecto y lo convierte a diapositivas. Cada vez que haces cambios, el contenido de las diapositivas se reflejará automáticamente. Por ejemplo:

```
Slides
```

Hola mundo

---

```
Página 2
```

Usa directamente bloques de código para el resultado

```
//```ts
console.log('¡Hola, mundo!')
//```
```

---

```
Página 3
```

Lee más sobre la sintaxis de Markdown de Slides en la [guía de sintaxis](#).

# Instalación

## Plantilla de inicio

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

La mejor manera de empezar es utilizando nuestra plantilla oficial de inicio.

Con NPM:

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

Con Yarn:

```
$ yarn create slidev
```

Sigue las indicaciones y se abrirá la presentación en <http://localhost:3030/> automáticamente para ti.

También contiene la configuración básica y una breve demostración con instrucciones sobre cómo empezar a utilizar Slidev.

## Instalación manual

Si todavía prefieres instalar Slidev manualmente o quieres integrarlo en tus proyectos existentes, puedes hacer lo siguiente:

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

```
$ touch slides.md
```

```
$ npx slidev
```

Por favor, nótese que si usas [pnpm](#), necesitarás activar la opción de '[shamefully-hoist](#)' para que Slidev funcione correctamente:

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

## Instalación global

Available since v0.14

Puedes instalar Sliddev globalmente con el siguiente comando:

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

Y luego usar `slidev` donde deseas sin que tengas que crear un proyecto cada vez.

```
$ slidev
```

Este comando también intentará utilizar el `@slidev/cli` local si se ha encontrado en `node_modules`.

## Instalación en Docker

Si necesitas una forma rápida de ejecutar una presentación con contenedores, puedes utilizar la imagen de Docker preconstruida y mantenida por [tangramor](#), o construir la tuya propia.

Simplemente ejecuta el siguiente comando en tu carpeta de trabajo:

```
docker run --name slidev --rm -it \
--user node \
-v ${PWD}:/slidev \
-p 3030:3030 \
tangramor/slidev:latest
```

Si tu carpeta de trabajo está vacía, generará una plantilla de `slides.md` y otros archivos relacionados en tu carpeta de trabajo, y lanzará el servidor en el puerto `3030`.

Puedes acceder a tus diapositivas desde `http://localhost:3030/`

## Construir una imagen desplegable

O puedes crear tu propio proyecto de Sliddev en Docker con Dockerfile:

```
FROM tangramor/slidev:latest
ADD . /slidev
```

Crea la imagen de Docker: `docker build -t mippt .`

Y ejecuta el contenedor: `docker run --name misdiapos --rm --user node -p 3030:3030 mippt`

Puedes visitar tus diapositivas desde `http://localhost:3030/`

## Constuir una SPA alojable (Single Page Application)

Ejecuta el comando `docker exec -i slidev npx slidev build` en el contenedor de `slidev` en ejecución. Generará archivos HTML estáticos en la carpeta `dist`.

### Alojar en Github Pages

Puedes alojar `dist` en un sitio web estático como [Github Pages](#) o Gitlab Pages.

Debido a que en las páginas de Github la URL puede contener subcarpetas, tienes que modificar el `index.html` generado y cambiar `href="/assets/xxx` por `href=". /assets/xxx`. O puedes utilizar la opción `--base=/<subcarpeta>/` durante el proceso de construcción, como por ejemplo `docker exec -i slidev npx slidev build --base=/slidev_docker/`.

Y para evitar el proceso de construcción de Jekyll, es necesario añadir un archivo vacío denominado `.no-jekyll`.

## Alojamiento con Docker

También puedes alojarlo por tu cuenta con Docker:

```
docker run --name myslides --rm -p 80:80 -v ${PWD}/dist:/usr/share/nginx/html
nginx:alpine
```

O crea una imagen estática con el siguiente Dockerfile:

```
FROM nginx:alpine
COPY dist /usr/share/nginx/html
```

Crea la imagen de Docker: `docker build -t mystaticppt .`

Y ejecuta el contendor: `docker run --name myslides --rm -p 80:80 mystaticppt`

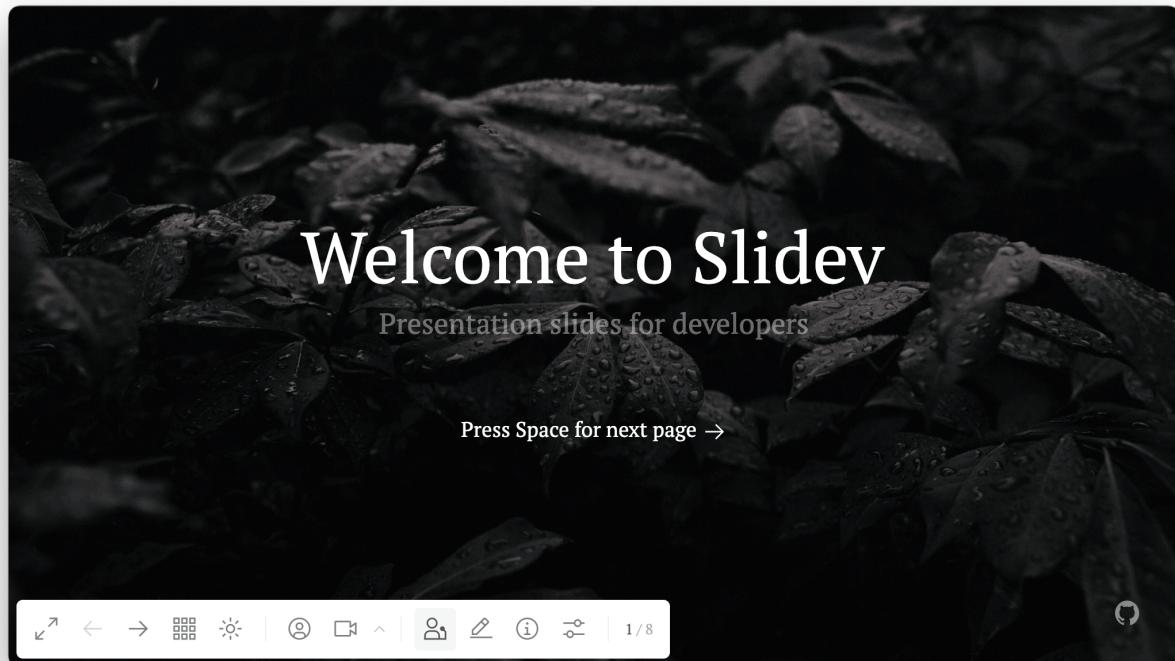
Puedes acceder a tus diapositivas desde `http://localhost/`

Echa un vistazo a [tangramor/slidev\\_docker](#) para más detalles.

# Navegación

## Barra de navegación

Mueve el ratón a la esquina inferior izquierda de la página de Slidev, aparecerá la barra de navegación.

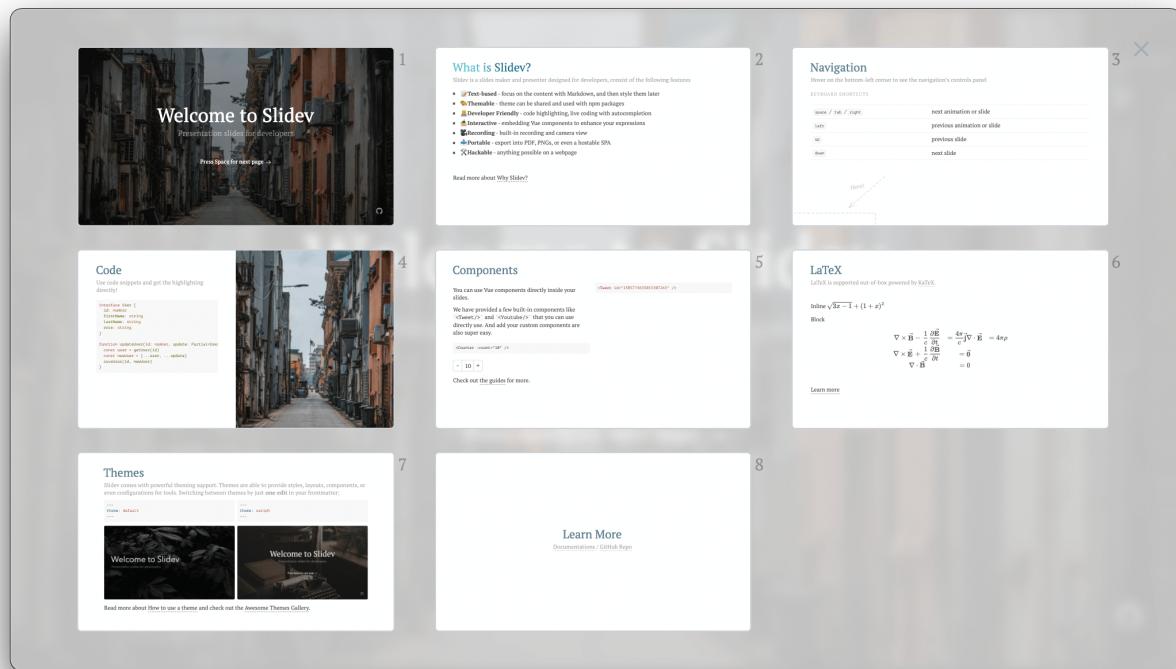


Atajos	Botón	Descripción
f		alternar pantalla completa
right / space		siguiente animación o diapositiva
left		animación o diapositiva anterior
up	-	diapositiva anterior
down	-	siguiente diapositiva
o		alternar el <a href="#">vistazo rápido de diapositivas</a>
d		alternar modo oscuro
-		alternar <a href="#">vista de cámara</a>
-		<a href="#">grabación</a>
-		entrar al <a href="#">modo presentador</a>
-		alternar el <a href="#">editor integrado</a>

Atajos	Botón	Descripción
-		descargar diapositivas (solo aparecen en la <a href="#">build SPA</a> )
-		mostrar información sobre las diapositivas
-		mostrar menú de opciones
g	-	mostrar ir a..

## Vistazo rápido de diapositivas

Pulsando o o haciendo clic en el botón de la barra de navegación, puedes tener un vistazo rápido de tus diapositivas para poder saltar entre ellas fácilmente.



[Go to TOC](#)

# Modo presentador

Haz clic en el botón del panel de navegación, o visita <http://localhost:3030/presenter> manualmente, para entrar en el modo presentador. Siempre que entres en el modo presentador, las demás instancias de la página se mantendrán automáticamente sincronizadas con el presentador.

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 heading "What is Sliddev?" which lists features like "Stream based", "Developer friendly", and "Recordable". 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 area, there are navigation icons for back, forward, search, and other presentation controls, along with a status bar showing "1 / 7".

---

[Go to TOC](#)

# Grabación

Slidev tiene integrados grabación y vista de cámara. Puedes usarlas para grabar tu presentación fácilmente en un solo lugar.

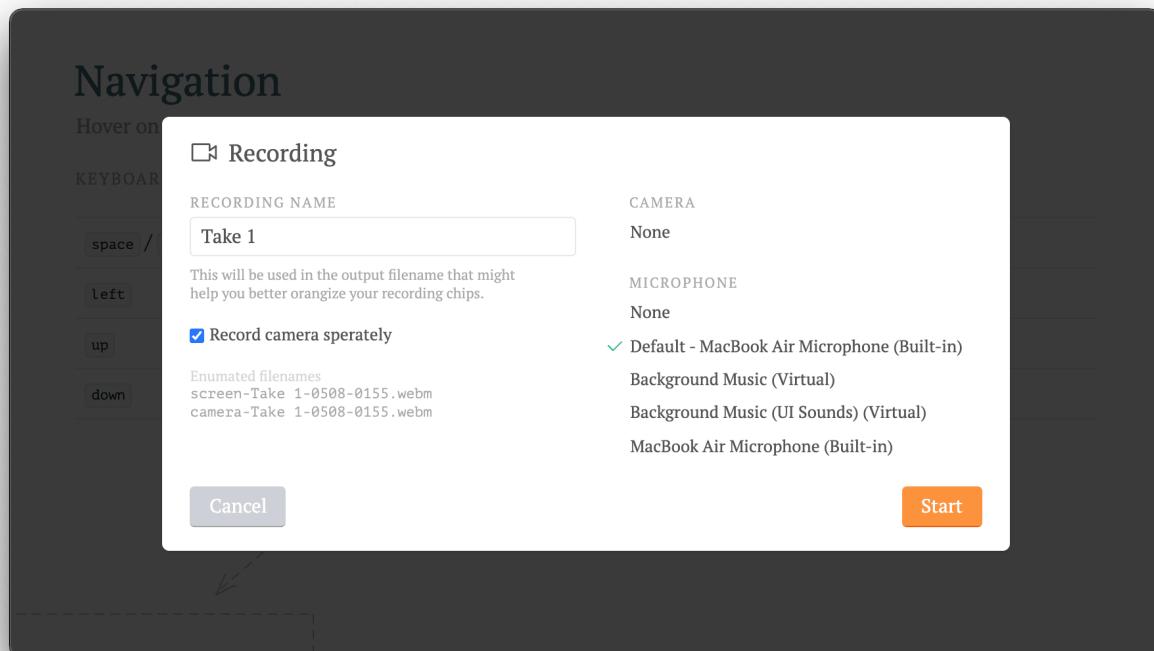
## Vista de cámara

Haz clic en el botón del panel de navegación para mostrar la vista de cámara en la presentación. Puedes arrastrarla para moverla y utilizar el manejador de la esquina inferior derecha para cambiar su tamaño. El tamaño y la posición persistirán en `localStorage` y por lo tanto serán consistentes a través de múltiples actualizaciones, así que no hay necesidad de preocuparse por eso.

## Grabación

Al hacer clic en el botón del panel de navegación, aparecerá un cuadro de diálogo. Aquí puedes elegir entre grabar tu cámara incrustada en tus diapositivas o separarlas en dos archivos de vídeo.

Esta función está impulsada por [RecordRTC](#) y utiliza la [WebRTC API](#).




---

[Go to TOC](#)

# Sintaxis de Markdown

Las diapositivas se escriben dentro de **un único archivo markdown** (por defecto `./slides.md`).

Puedes utilizar [las características de Markdown](#) como lo harías normalmente, con soporte adicional de HTML en línea y componentes de Vue. También se admite el estilo utilizando [Windi CSS](#). Utiliza `---` separado con una nueva línea para dividir tus diapositivas.

```
Slides
```

¡Hola, mundo!

```

```

```
Página 2
```

Usa directamente bloques de código para el resultado

```
//```ts
console.log('¡Hola, mundo!')
//```

```

```
Página 3
```

Puedes utilizar directamente Windi CSS y los componentes de Vue para estilizar y enriquecer tus diapositivas.

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

## Front Matter y plantillas

Puedes especificar plantillas y otros metadatos para cada diapositiva convirtiendo los separadores en [bloques de Front Matter](#). Cada bloque de entrada comienza con un guión triple y termina con otro. Los textos entre ellos son objetos de datos en formato [YAML](#). Por ejemplo:

```

layout: cover

Slides

Esta es la página de portada

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

Página 2
```

Esta es una página con la plantilla `centrado` y una imagen de fondo.

---

# Página 3

Esta es la página por defecto sin ningún metadato adicional.

Revisa la [página de personalización](#) para más detalles.

## Bloques de código

Una gran razón por la que estoy construyendo Slidev es la necesidad de hacer que mi código se vea bien en las diapositivas. Así que, tal y como esperabas, puedes utilizar el bloque de código de estilo Markdown para resaltar tu código.

```
//``ts
console.log('Hello, World!')
//``
```

Damos soporte a [Prism](#) y [Shiki](#) como resaltadores de sintaxis. Revisa la [sección de resaltadores](#) para más detalles.

## Resaltado de línea

Para resaltar líneas específicas, simplemente añade los números de línea dentro del paréntesis `{}`. Los números de línea empiezan a contar desde 1.

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

Para cambiar el resultado en varios pasos, puede utilizar `|` para separarlos. Por ejemplo

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

Esto resaltará primero `a: Ref<number> | number` y `b: Ref<number> | number`, y después `return computed(() => unref(a) + unref(b))` tras un clic, y finalmente, todo el bloque. Aprende más en la [guía de animaciones de clics](#).

## Editor de Monaco

Cuando quieras hacer alguna modificación en la presentación, simplemente añade `{monaco}` después del identificador del lenguaje - ise convierte todo el bloque en un editor de Monaco con todas las funciones!

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

Lea más sobre [configurar Monaco](#).

## Estilos incrustados

Puedes usar la etiqueta `<style>` en tu Markdown directamente para sobreescibir los estilos de la **diapositiva actual**.

```
Esto es rojo

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

La siguiente diapositiva no se ve afectada
```

La etiqueta `<style>` en Markdown siempre está [focalizada](#). Para reemplazar los estilos globales, echa un vistazo a la [sección de personalización](#)

Gracias a [Windi CSS](#), puedes usar CSS anidado y [directivas](#) (por `@apply`)

```
Slides

> Hello `world`

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

## Recursos estáticos

Al igual que harías en markdown, puedes utilizar imágenes que apunten a una url remota o local.

Para los recursos remotos, `vite-plugin-remote-assets` los almacenará en caché tras la primera ejecución para que puedan tener una carga instantánea incluso para las imágenes más pesadas.

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

Para recursos locales, muévelos a la carpeta `public` y referéncialos con una **barrita** ( / ).

```
! [Imagen local](pic.png)
```

Si deseas aplicar tamaños o estilos personalizados, puedes convertirlos a la etiqueta `<img>`:

```

```

## Notas

También puedes tomar notas para cada diapositiva. Se mostrarán en el [Modo Presentador](#) para que puedas consultarlas durante las presentaciones.

En Markdown, el último bloque de comentarios de cada diapositiva será tratado como una nota.

```

layout: cover

Página 1

Esta es la página de la portada.

<!-- Esto es una nota -->

Página 2

<!-- Esto NO es una nota porque precede al contenido de la diapositiva -->

La segunda página

<!--
Esta es otra nota
-->
```

## Iconos

Slidev te permite tener acceso a casi todos los conjuntos de iconos populares de código abierto **directamente** en tu markdown. Potenciado por `vite-plugin-icons` e [Iconify](#).

El nombrado sigue la convención de [Iconify](#): `{collection-name}-{icon-name}`. Por ejemplo:

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

Puedes navegar y buscar todos los iconos con [Icônes](#).

## Estilizando iconos

Puedes estilizar los iconos tal y como estilizarías un elemento HTML. Por ejemplo:

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

## Slots

Disponible desde v0.18

Algunas plantillas pueden proporcionar múltiples puntos de contribución utilizando los `<slots>` nombrados de Vue.

Por ejemplo, en una estructura de dos columnas, puede tener dos columnas a la izquierda (slot `default`) y a la derecha (slot `right`) una al lado de la otra.

```

layout: two-cols

<template v-slot:default>
Izquierda
 Esto muestra a la izquierda
</template>
<template v-slot:right>
Derecha
 Esto se muestra a la derecha
</template>
```

### Izquierda

Esto se muestra a la izquierda

### Derecha

Esto se muestra a la derecha

También proporcionamos una sintaxis abreviada (`::nombre::`) para el nombre del slot. El siguiente ejemplo funciona exactamente igual que el anterior.

```

layout: two-cols

```

```
Izquierda
Esto muestra a la izquierda
::right::
Derecha
Esto se muestra a la derecha
```

También puede especificar explícitamente el slot por defecto y proporcionar un orden personalizado

```

layout: two-cols

::right::

Derecha
Esto se muestra a la derecha
::default::

Izquierda
Esto muestra a la izquierda
```

## Configuraciones

Todas las configuraciones necesarias se pueden definir en el archivo Markdown. Por ejemplo:

```

theme: serif
layout: portada
background: 'https://source.unsplash.com/1600x900/?nature,water'

Slidelv
Esta es la página de la portada
```

[Lee más sobre la configuración de Front Matter](#)

## LaTeX

Slidev viene con soporte integrado de LaTex, gracias a [KaTeX](#).

### En líneas

Rodea tu LaTeX con un solo `$` a cada lado para la representación en línea.

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

## Bloque

Utiliza dos ( `$$` ) para la representación de bloques. Este modo utiliza símbolos más grandes y centra el resultado.

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

Aprende más: [Demo](#) | [KaTeX](#) | [markdown-it-katex](#)

## Diagramas

También puedes crear diagramas / gráficos a partir de descripciones textuales en tu Markdown, gracias a [Mermaid](#).

Los bloques de código marcados como `mermaid` se convertirán en diagramas, por ejemplo:

```
//```mermaid
sequenceDiagram
 Alice->John: Hello John, how are you?
 Note over Alice,John: A typical interaction
//````
```

Además, puedes pasarle un objeto de opciones para especificar la escala y el tema. La sintaxis del objeto es un literal de objeto de JavaScript, tendrás que añadir comillas ( `'` ) para las cadenas y utilizar comas ( `,` ) entre las claves.

```
//```mermaid {theme: 'neutral', scale: 0.8}
graph TD
 B[Text] --> C{Decision}
 C -->|One| D[Result 1]
 C -->|Two| E[Result 2]
//````
```

Aprende más: [Demo](#) | [Mermaid](#)

# Entradas múltiples

Disponible desde v0.15

Puedes dividir tu `slides.md` en varios archivos y organizarlos como quieras.

`slides.md` :

```
Página 1
Esta es una página normal

src: ./subpagina2.md

<!-- esta página se cargará desde './subpagina2.md' -->
El contenido en línea será ignorado
```

`subpagina2.md` :

```
Página 2
Esta página es de otro archivo
```

## Fusión de Front Matter

Puedes proporcionar *Front Matters* tanto de tu entrada principal como de páginas externas de markdown. Si hay las mismas claves en ellas, las de la **entrada principal tienen mayor prioridad**. Por ejemplo:

`slides.md` :

```

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

```

`cover.md` :

```

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

Portada
Página de portada
```

Acabarán siendo equivalentes a la página siguiente:

```

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

Portada

Página de portada
```

## Reutilización de páginas

Con el soporte de entradas múltiples, la reutilización de páginas puede ser sencilla. Por ejemplo:

```

src: ./cover.md

src: ./intro.md

src: ./content.md

reutilización
src: ./content.md

```

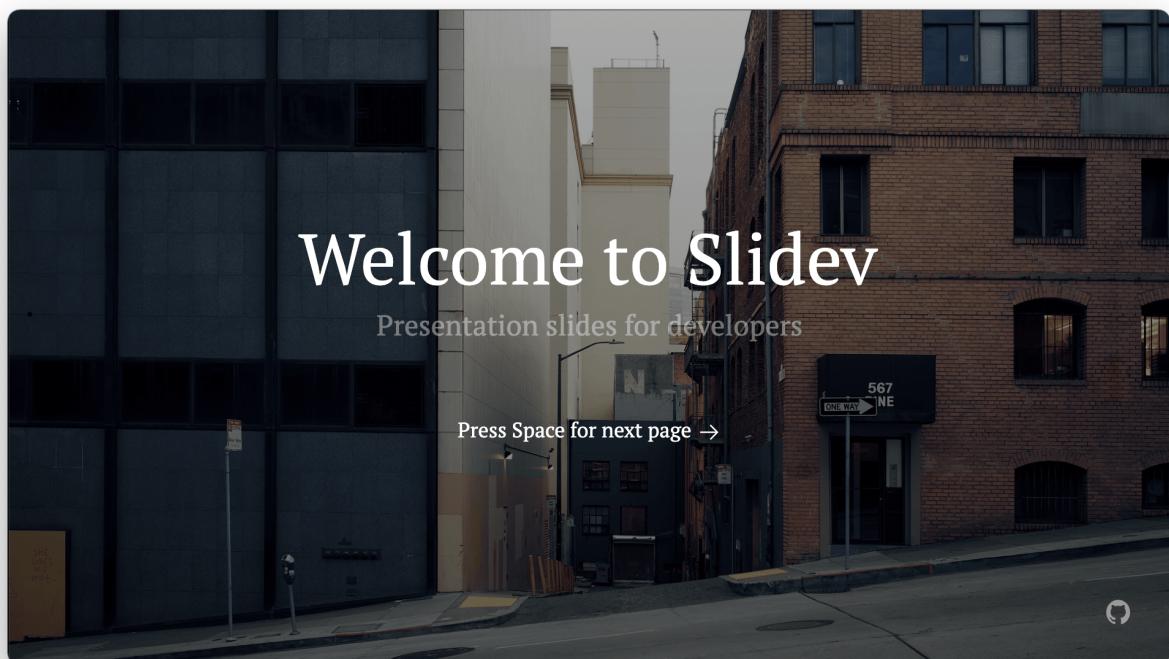
# Por qué Slidev

Hay muchos creadores de diapositivas *WYSIWYG* de uso general y repletos de funciones, como [Microsoft PowerPoint](#) y [Apple Keynote](#). Funcionan bastante bien para hacer bonitas diapositivas con animaciones, gráficos y muchas otras cosas, a la vez que son muy intuitivas y fáciles de aprender. Entonces, ¿por qué molestarse en hacer Slidev?

Slidev pretende proporcionar la flexibilidad y la interactividad para que los desarrolladores puedan hacer sus presentaciones aún más interesantes, expresivas y atractivas utilizando las herramientas y tecnologías con las que ya están familiarizados.

Cuando se trabaja con editores *WYSIWYG*, es fácil distraerse con las opciones de estilo. Slidev lo remedia separando el contenido y lo visual. Esto le permite centrarse en una cosa a la vez, a la vez que puede reutilizar los temas de la comunidad. Slidev no pretende sustituir por completo a otros creadores de presentaciones de diapositivas. Más bien, se centra en atender a la comunidad de desarrolladores.

## Slidev



Estas son algunas de las características más interesantes de Slidev:

## Basado en Markdown

Slidev utiliza un formato Markdown extendido para almacenar y organizar tus diapositivas en un único archivo de texto. Esto te permite centrarte en la elaboración del contenido. Y como el contenido y los estilos están separados, esto también hace posible cambiar entre diferentes temas sin esfuerzo.

Aprende más sobre la [sintaxis de Markdown de Slidev](#). ## Tematizable

Los temas para Slidev pueden ser compartidos e instalados usando módulos de npm. Luego se aplican con una sola línea de configuración.

Revisa la [galería de temas](#) o [aprende cómo escribir un tema](#).

## Pensado para desarrolladores

Slidev proporciona soporte de primera clase de los fragmentos de código para los desarrolladores. [Prism](#) y [Shiki](#) consiguen resultado de sintaxis perfecto, al tiempo que es capaz de editar el código en cualquier momento. Con el [editor de Monaco](#) integrado, también te permite programar en directo en tu presentación con autocompletado, marcado de tipos e incluso soporte de TypeScript.

Lee más sobre los [resaltadores](#) y la [configuración de Monaco](#).

## Rápido

Slidev es potenciado por [Vite](#), [Vue 3](#) y [Windi CSS](#), lo que da la mejor experiencia de editor. Cada cambio que hagas se reflejará en las diapositivas **instantáneamente**.

Aprende más sobre [nuestras tecnologías](#)

## Interactivo y expresivo

Puedes escribir componentes Vue personalizados y utilizarlos directamente dentro de tu archivo markdown. También puedes interactuar con ellos dentro de la presentación para expresar tu idea de una manera más interesante e intuitiva.

Aprende más sobre [grabar aquí](#).

## Portable

Exporta sus diapositivas a PDF, PNG o incluso a una *Single Page Application* (SPA) alojable con un solo comando, y compártalas en cualquier lugar.

Lee más sobre esto en [exportando documentos](#).

## Hackable

Al estar basado en tecnologías web, todo lo que se puede hacer en una aplicación web también es posible con Slidev. Por ejemplo, WebGL, solicitudes de API, iframes, o incluso compartir en vivo. ¡Todo depende de tu imaginación!

## Pruébalo

Trastear con Slidev te dirá más que mil palabras. Estás tan solo a un comando:

```
$ npm init slidev
```

O echa un vistazo rápido:

Slidev First Preview Demo

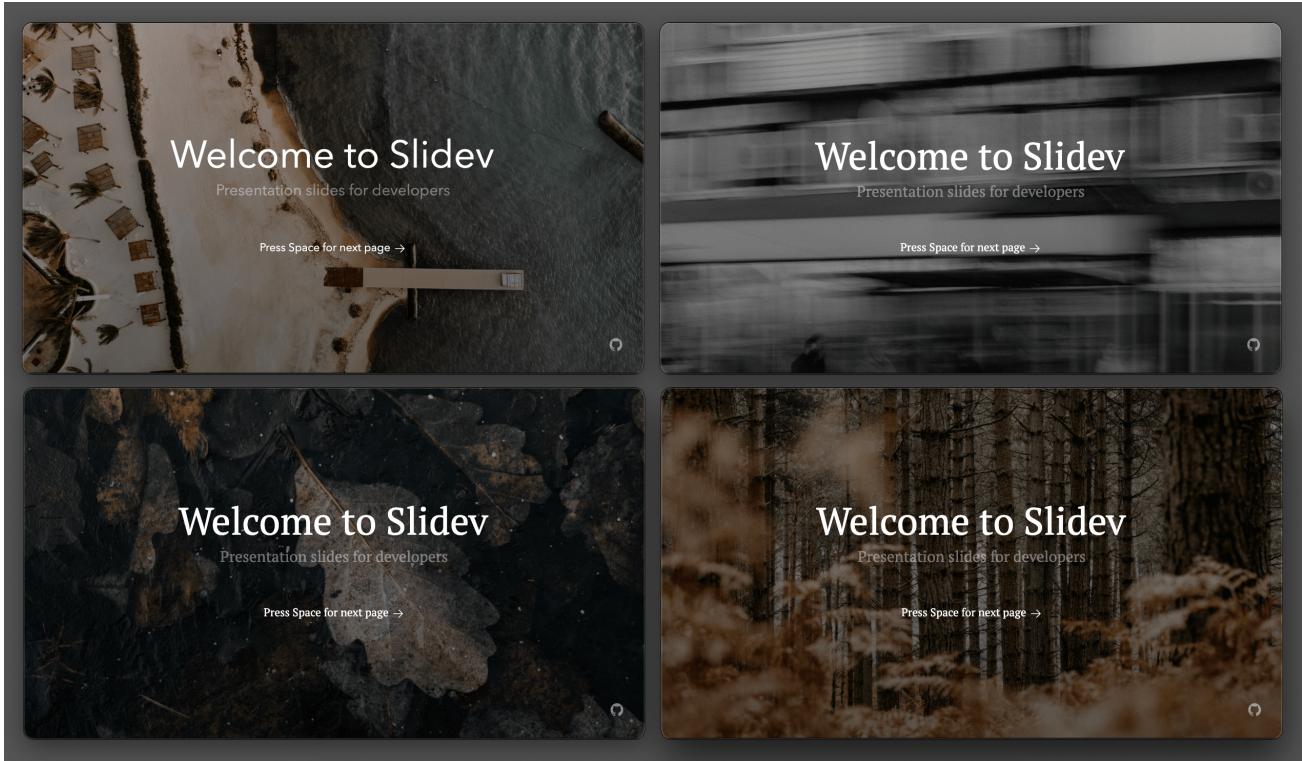


---

[Go to TOC](#)

# Portadas seleccionadas

Hemos seleccionado algunas imágenes de portada para nuestras plantillas de inicio:



```

random image from the curated collection
background: https://source.unsplash.com/collection/94734566/1920x1080

```

Si te gusta alguna de ellas, echa un vistazo a nuestra [colección Unsplash](#) y descubre sus autores.

# Recursos de aprendizaje

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

## 日本語

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

---

[Go to TOC](#)

# Galería

Charlas / Presentaciones usando Slides:

---

[Go to TOC](#)

# Galería de temas

Navega por los geniales temas disponibles para Slidev en esta página.

iLee más sobre [cómo usar un tema](#), o [cómo escribir el tuyo propio](#) y compártelo con la comunidad!

## Temas oficiales

## Temas de la comunidad

Aquí están los temas elaborados por la comunidad.

## Más temas

Encuentra todos los [temas disponibles en NPM](#).

---

[Go to TOC](#)

# Usar un tema

Cambiar el tema en Slidev es sorprendentemente fácil. Todo lo que necesitas hacer es añadir el campo `theme:` en tu frontmatter.

```

theme: serif

```

Puedes iniciar el servidor, que te preguntará instalar el tema automáticamente

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

o instalar el tema manualmente a través de

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

Y eso es todo, disfruta del nuevo tema! Para más detalles sobre el uso, puedes consultar el *README* del tema.

¿Quieres compartir tu tema? Aprende sobre [cómo escribir un tema](#).

## Ejectar tema

Si quieres tener el control total del tema actual, puedes usar **eject** en tu sistema de archivos local y modificarlo como quieras. Ejecutando el siguiente comando:

```
$ slidev theme eject
```

Ejectará el tema que está usando actualmente en `./theme`, y cambiará su frontmatter a

```

theme: ./theme

```

Esto también podría ser útil si quieres hacer un tema basado en uno existente. Si lo haces, recuerda mencionar el tema original y al autor :)

## Tema local

Como probablemente descubriste en la sección anterior, puedes tener un tema local para tu proyecto. Teniendo una **ruta relativa** en su campo de tema:

```

theme: ./path/to/theme

```

Revisa [cómo escribir un tema](#) para más detalles.

---

[Go to TOC](#)

# Escribir un tema

Para empezar, te recomendamos que utilices nuestro generador para armar tu primer tema

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

Luego puedes modificar y jugar con él. También puede consultar los [temas oficiales](#) como ejemplos.

## Capacidad

Un tema puede contribuir a los siguientes puntos:

- Estilos globales
- Provide configuraciones por defecto (fuentes, esquema de colores, resaltadores, etc.)
- Proporcionar diseños personalizados o anular los existentes
- Proporcionar componentes personalizados o anular los existentes
- Extender las configuraciones de Windi CSS
- Configurar herramientas como Monaco y Prism
- Provide default configurations (fonts, color schema, highlighters, etc.)
- Provide custom layouts or override the existing one
- Provide custom components or override the existing one

## Convenciones

Los temas se publican en el registro npm, y deben seguir las siguientes convenciones:

- El nombre del paquete debe empezar por `slidev-theme-`, por ejemplo: `slidev-theme-awesome`.
- Añade `slidev-theme` y `slidev` en el campo `keywords` de tu `package.json`.

## Configuración

Para configurar el campo de pruebas para su tema, puede crear `example.md` con el siguiente frontmatter, para decirle a Slidev que estás usando el directorio actual como un tema.

```

theme: ../

```

Opcionalmente, también puedes añadir algunos scripts a tu `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 tu tema, simplemente ejecuta `npm publish` y estarás listo. No se requiere ningún proceso de transpilación (lo que significa que puedes publicar directamente los archivos `.vue` y `.ts`, Slidev es lo suficientemente inteligente como para entenderlos).

Los puntos de contribución del tema siguen las mismas convenciones que la personalización local, por favor consulta [los documentos para las convenciones de nomenclatura](#).

## Configuraciones por defecto

Disponible desde v0.19

Un tema puede proporcionar por defecto [configuraciones](#) vía `package.json`.

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

Las fuentes se importarán automáticamente desde [Google Fonts](#).

Aprende más sobre [las fuentes](#) y [configuraciones de front matter](#)

## Metadatos del tema

### Esquema de colores

Por defecto, Slidev asume que los temas soportan tanto el modo claro como el modo oscuro. Si sólo quieres que tu tema se presente en un esquema de color diseñado, tendrás que especificarlo explícitamente en `package.json`.

```
// package.json
{
 "name": "slidev-theme-my-cool-theme",
 "keywords": [
 "slidev-theme",
 "slidev"
],
 "slidev": {
 "colorSchema": "light" // o "dark" o "both"
 }
}
```

Para acceder al modo oscuro al crear tus estilos de tema, puedes envolver el css específico del modo oscuro dentro de una clase `dark`:

```
/* css general aquí */

html:not(.dark) {
 /* css del modo claro aquí */
}

html.dark {
 /* css del modo oscuro aquí */
}
```

Slidev alterna una clase `dark` en el elemento `html` de la página para cambiar el esquema de color.

## Resaltador

Los colores de resaltado de sintaxis también se proporcionan en el tema. Soportamos tanto [Prism](#) como [Shiki](#). Para más información, consulte [los documentos de resaltado de sintaxis](#).

Puedes soportar uno de ellos, o ambos. Consulta el tema por defecto para los ejemplos de configuración `./styles/prism.css` / `./setup/shiki.ts`.

Además, recuerda especificar los resaltadores compatibles en su `package.json`.

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

## Versión de Slidev

Si el tema depende de una característica específica de Slidev que se ha introducido recientemente, puedes establecer la versión mínima de Slidev necesaria para que tu tema funcione correctamente:

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

Si los usuarios utilizan versiones anteriores de Slidev, se producirá un error.

---

[Go to TOC](#)

# Colophon

This book is created by using the following sources:

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