

Estándar de Frontend

Cuando eres parte de un equipo que aún no ha establecido reglas a seguir, es muy común que los desarrolladores escriban código a su modo, dando como resultado un repositorio difícil de entender y por tanto difícil de mantener.

También puede propiciar que ciertos desarrolladores escriban código con su propio estilo y otros lo hacen diferente. Al momento de unificar el trabajo hay discusiones para elegir la mejor opción, y al no tener reglas establecidas u oficialmente validadas por todo el equipo, la discusión puede ser más larga de lo normal, provocando retrasos en la entrega de nuevas implementaciones.

Definir un estándar de desarrollo en un proyecto de software es crucial por varias razones:

- **Calidad del producto final:** La implementación de ciertas reglas en el desarrollo de software mejora significativamente la calidad del producto final.
- **Mayor confiabilidad y mantenibilidad:** Los estándares de calidad de software forman parte de la ingeniería de software, y su utilización en el diseño, programación, prueba y análisis del software desarrollado ofrece una mayor confiabilidad y mantenibilidad.
- **Mejora continua:** Los estándares proporcionan un marco de referencia para evaluar la efectividad de los procesos actuales, facilitando la definición de actividades, prioridades y metas para garantizar la mejora continua.
- **Uniformidad en el código:** Establecer un estándar de código único facilita la comprensión y colaboración entre los desarrolladores.
- **Adaptabilidad a cualquier proyecto:** Los estándares están pensados de forma genérica para cualquier tipo de proyecto en cualquier industria.

Por lo tanto, es muy importante definir un estándar de desarrollo al inicio de un proyecto de software. Esto ayudará a garantizar que el equipo esté en la misma página y que el producto final sea de calidad.

Habilidades técnicas

- **Prototipar en FIGMA**

Los estudiantes sepan utilizar la herramienta para elaborar paleta de colores, mockups, un diseño responsivo y aplicar las técnicas de Wireframing.

- **Dominio de HTML/CSS:**

Los estudiantes deben contar con una base sólida de HTML para estructurar el contenido y CSS para los estilos.

- **Dominio de JavaScript:**

Comprensión sólida de los conceptos básicos de JavaScript, incluidas las variables, los tipos de datos, las funciones y las estructuras de control.

- **Frameworks y librerías:**

Conocer los marcos y bibliotecas de front-end populares como React, Angular o Vue.js, según la demanda y las tendencias de la industria.

Conocimiento acerca de frameworks de componentes como tailwind CSS, Bootstrap.

- **Diseño Responsivo:**

Conocer los principios del diseño web responsive, incluidos los media queries, grid y flexbox.

- **Preprocesadores CSS:**

Familiarizar a los estudiantes con los preprocesadores de CSS como Sass o Less para mejorar la capacidad de mantenimiento y reutilización de los estilos.

- **Control de versiones:**

Conocer los conceptos básicos de los sistemas de control de versiones, como Git, y los flujos de trabajo colaborativos utilizando plataformas como GitHub o GitLab y buenas prácticas de Git Flow o Trunk based Development.

Guia de diseño

Define un conjunto de principios, pautas y mejores prácticas destinadas a orientar el proceso de diseño y desarrollo de la interfaz de usuario (UI) y la experiencia del usuario (UX) de una aplicación. Estos aspectos incluyen los colores, los estilos, la tipografía, los iconos, la estructura y la navegación.

Estas guías de diseño tienen varios objetivos y beneficios:

- **Consistencia visual y de interacción:** Una guía de diseño define un conjunto coherente de elementos visuales y de interacción que deben aplicarse en toda la aplicación. Esto asegura que los usuarios experimenten una interfaz consistente, lo que facilita la comprensión y el uso de la aplicación.
- **Eficiencia en el desarrollo:** Al seguir una guía de diseño, los desarrolladores pueden trabajar de manera más eficiente al tener un conjunto de directrices ya establecido. Esto reduce la duplicación de esfuerzos y facilita la colaboración en equipos de desarrollo.
- **Mejora la usabilidad:** Al enfocarse en la experiencia del usuario, las guías de diseño buscan mejorar la usabilidad de la aplicación. Esto implica asegurar que la navegación sea intuitiva, que las funciones sean accesibles y que la interfaz sea fácil de entender.
- **Adaptabilidad y escalabilidad:** Esto implica crear un diseño que funcione bien en diferentes dispositivos y tamaños de pantalla, así como ser capaz de crecer y evolucionar a medida que la aplicación crece.
- **Facilita el mantenimiento:** Un conjunto coherente de directrices de diseño facilita el mantenimiento a lo largo del tiempo. Los equipos de desarrollo pueden realizar actualizaciones y mejoras de manera más eficiente si siguen un conjunto establecido de principios de diseño.
- **Alineación con objetivos del negocio:** Las guías de diseño también deben alinearse con los objetivos y la identidad de la marca del negocio. Esto garantiza que la interfaz de usuario refleje la visión y los valores de la empresa.

What's Material?

Material Design is a design system built and supported by Google designers and developers. Material.io includes in-depth UX guidance and UI component implementations for Android, Flutter, and the Web.

The latest version, Material 3, enables personal, adaptive, and expressive experiences – from dynamic color and enhanced accessibility, to foundations for large screen layouts and design tokens.

- UX foundations**
Foundations like color, type, and shape are customizable systems in Material
- Open-source code**
Multi-platform code to build beautiful products, faster
- Tutorials, case studies & news**

Fluent 2

WELCOME TO

Fluent 2

Let the *{ideas}* flow

[Check out what's new](#)

[Go to Fluent 1](#)

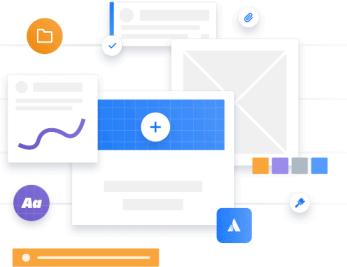
Design
Dive into our Figma UI kits

Develop
Find installation guides and tooling tips

Design, develop, deliver

Use Atlassian's end-to-end design language to create simple, intuitive and beautiful experiences.

[Get started](#)



shadcn/ui Docs Components Themes Examples GitHub [Search documentation...](#)

Inbox		128
Drafts	9	
Sent		
Junk	23	
Trash		
Archive		
Social	972	
Updates	342	
Forums	128	
Shopping	8	
Promotions	21	

Inbox

All mail Unread

Search

William Smith 3 months ago
Meeting Tomorrow
Hi, let's have a meeting tomorrow to discuss the project. I've been reviewing the project details and have some ideas I'd like to share. It's crucial that we align on our...

Alice Smith 3 months ago
Re: Project Update
Thank you for the project update. It looks great! I've gone through the report, and the progress is impressive. The team has done a fantastic job, and I appreciate the hard...

Bob Johnson 10 months ago
Weekend Plans
Any plans for the weekend? I was thinking of going hiking in the nearby mountains. It's been a while since we had some outdoor fun. If you're interested, let me know, an...

Emily Davis 10 months ago
Re: Question about Budget
I have a question about the budget for the upcoming project. It seems like there's a discrepancy in the allocation of resources. I've reviewed the budget report and...

Diseño y flujos en Figma

Figma ofrece varias ventajas que contribuyen a un proceso de desarrollo más eficiente y colaborativo. Además de poder crear flujos para definir cómo va a interactuar el usuario con nuestra aplicación sin haber desarrollado la aplicación funcional como tal.

Razones para usar Figma como herramienta de trabajo:

- **Colaboración en tiempo real:** Figma permite a múltiples usuarios trabajar en un diseño al mismo tiempo, lo que facilita la colaboración en equipos distribuidos. Puedes ver las actualizaciones en tiempo real y realizar comentarios directamente en el diseño.
- **Accesibilidad desde cualquier lugar:** Al ser una herramienta basada en la nube, puedes acceder a tus diseños desde cualquier dispositivo con conexión a Internet. Esto facilita el trabajo remoto y la colaboración entre equipos ubicados en diferentes lugares.
- **Prototipado interactivo:** Se pueden crear transiciones, animaciones y enlaces entre diferentes pantallas para simular la experiencia de usuario final.
- **Componentes reutilizables:** Permite crear componentes reutilizables, lo que facilita la consistencia en el diseño. Si actualizas un componente, se actualizará automáticamente en todas las instancias, ahorrando tiempo y asegurando la coherencia en todo el diseño.
- **Librerías compartidas:** Permite crear bibliotecas de diseño compartidas que contienen elementos de diseño reutilizables, como botones, iconos y estilos de texto. Esto ayuda a mantener la consistencia en todos los proyectos y facilita la actualización de elementos comunes.

Uso de framework frontend

Los frameworks facilitan el desarrollo de aplicaciones al proporcionar funciones y componentes, así como una guía para la arquitectura de la aplicación, esto ayuda a los desarrolladores a ahorrar tiempo y esfuerzo, y a crear aplicaciones que sean consistentes y fáciles de usar.

Framework	Características principales
React	Basado en componentes, rendimiento excepcional, comunidad activa
Vue.js	Basado en componentes, progresivo, fácil de aprender
Angular	Completo, maduro, potente, curva de aprendizaje alta
Svelte	Ligero, rendimiento, curva de aprendizaje suave

Estructura de carpetas

Módulo Core: Incluyen servicios globales que se usarán en toda la aplicación.

Módulo Shared: Incluyen componentes, servicios, utilitarios, funciones que se utilizaran en múltiples partes de la aplicación.

Módulo Layout: Incluyen los contenedores y componentes que arman la estructura base de la aplicación.

Módulo de features: Incluyen vistas, componentes, funcionalidades dedicadas a un dominio de la aplicación.

```
/src
  └── app
    ├── core
    │   ├── services
    │   │   ├── AuthService
    │   │   └── ...
    │   ├── interceptors
    │   │   ├── AuthInterceptor
    │   │   └── ...
    │   ├── guards
    │   │   ├── AuthGuard
    │   │   └── ...
    │   ├── interfaces
    │   │   ├── User
    │   │   └── ...
    ├── feature
    │   ├── users
    │   ├── products
    │   └── orders
    ├── layout
    │   └── admin
    │       ├── admin.component.ts
    │       └── admin.component.html
    └── shared
        ├── components
        │   ├── SpinnerComponent
        │   ├── AlertComponent
        │   └── ...
        ├── directives
        │   ├── HighlightDirective
        │   └── ...
        └── utils
            ├── helpers.ts
            └── constants.ts
```

Uso de Typescript

Razones para usar Typescript en proyectos web:

- **Tipado estático:** Proporciona un sistema de tipos estáticos que ayuda a detectar errores en tiempo de compilación.
- **Mejora la legibilidad:** Al definir tipos para variables y funciones, el código se vuelve más claro y fácil de entender, facilitando la colaboración y el mantenimiento a medida que el proyecto crece.
- **Autocompletado y sugerencias:** Ofrece un soporte sólido para autocompletado en los editores de código, aumentando la productividad al proporcionar sugerencias precisas durante la escritura del código.
- **Refactorización segura:** Facilita operaciones de refactorización al permitir cambios precisos y seguros, gracias a la información de tipos que el sistema proporciona.
- **Integración con herramientas de desarrollo:** Se integra de manera efectiva con herramientas populares de desarrollo, como linters y pruebas unitarias, mejorando la calidad del código y facilitando la adopción de buenas prácticas.
- **Compatibilidad con ECMAScript:** Permite aprovechar las últimas características de JavaScript, incluso antes de que estén completamente soportadas por los navegadores, y ofrece opciones de compilación para diferentes versiones de ECMAScript.
- **Escalabilidad del código:** Ayuda a gestionar la complejidad a medida que el código base crece, proporcionando una base sólida para proyectos grandes y equipos de desarrollo.
- **Comunidad y soporte:** Cuenta con una comunidad activa y el respaldo de Microsoft, lo que significa un amplio soporte, recursos y herramientas disponibles para los desarrolladores.

Uso de Index.ts

La práctica de definir un archivo index.ts para las exportaciones es una convención que ofrece varios beneficios en términos de organización y facilita el consumo de módulos.

- **Simplificación del consumo de módulos:** Puedes exportar todas las funcionalidades importantes desde ese directorio en un solo lugar. Y los consumidores del módulo pueden importar fácilmente todas las exportaciones usando una única importación, lo que hace que el código sea más limpio y legible.
- **Mejora de la legibilidad y mantenibilidad:** Facilita la navegación y comprensión de la estructura del módulo, especialmente en proyectos más grandes.

Variables de entorno

Usar variables de entorno es una práctica común en el desarrollo de software, aquí algunas de las razones:

- **Configuración dinámica:** Las variables de entorno permiten configurar dinámicamente aspectos del comportamiento de tu aplicación o servicio. En lugar de codificar configuraciones directamente en el código, puedes ajustar el comportamiento de la aplicación sin necesidad de recompilar o modificar el código fuente.
- **Seguridad:** Almacenar información sensible, como claves de API, contraseñas u otros datos confidenciales. Esto reduce el riesgo de exposición accidental de información sensible si el código fuente se comparte públicamente o si el repositorio es accesible a personas no autorizadas.
- **Ambientes múltiples:** Las variables de entorno son especialmente útiles al trabajar con diferentes entornos (desarrollo, pruebas, producción, etc.). Puedes configurar fácilmente diferentes valores para las variables de entorno en cada entorno

Reglas Lint

Estas reglas contribuyen a mantener la calidad, mantenibilidad y consistencia del código.

Razones clave para utilizar reglas de linting:

- **Detección temprana de errores:** Las reglas de linting ayudan a detectar errores y posibles problemas en el código durante el desarrollo.
- **Consistencia del código:** Las reglas de linting permiten establecer y mantener un estilo de codificación consistente en todo el proyecto.
- **Alineación con mejores prácticas:** Las herramientas de linting a menudo incluyen reglas que promueven el uso de mejores prácticas de programación.
- **Facilita el trabajo en equipo:** Al adoptar reglas de linting, se establece un conjunto común de normas para el equipo de desarrollo. Esto reduce la fricción y mejora la colaboración al garantizar que todos los desarrolladores estén en la misma página en cuanto a la estructura y el estilo del código.
- **Facilita la incorporación de nuevos miembros:** Los nuevos desarrolladores pueden adaptarse más rápidamente al estilo y las convenciones del código existente.

Eslint for angular: [@angular-eslint/eslint-plugin](#)

ESLint Rules: [Rules Reference - ESLint](#)

TSConfig rules: [TypeScript: Referencia TSConfig](#)