

The cover features decorative geometric shapes in the top-left and bottom-right corners. These shapes are composed of overlapping triangles and parallelograms in two colors: a dark navy blue and a light blue-grey. The shapes are arranged in a way that creates a sense of movement and depth, framing the central text.

INFORME DE PROYECTO

Javier Herrera
Estandarizaciones y Recomendaciones

Estandarizaciones de Commits en GitHub

Las estandarizaciones de commits en GitHub son prácticas recomendadas que buscan uniformar el formato y el contenido de los mensajes de commit. Estas prácticas facilitan la lectura y comprensión del historial de cambios en un proyecto, mejorando así la colaboración y el mantenimiento del código.

Estructura Básica de un Commit:

<Naturaleza del commit>(<\$Área donde afecta>): <Mensaje/Descripción del cambio>

Ejemplos:

- fix(bug): arreglo en la distribución de carpetas
- docs(add): se agregó HTML, CSS y JavaScript

Nota: No debe haber nada en mayúsculas dentro del propio commit, excepto cuando la mayúscula no es al inicio del mensaje/descripción del cambio.

Naturaleza del Commit

La naturaleza del commit se divide en varios campos. Los más utilizados y su propósito son los siguientes:

- build: Cambios en el sistema de compilación o en dependencias externas.
- ci: Cambios en las configuraciones CI y scripts.
- docs: Cambios en la documentación.
- feat: Agregación, eliminación o cambio de una característica.
- fix: Corrección de algún bug dentro del programa.
- perf: Mejora del rendimiento de la app.
- refactor: Cambios que no corrigen un error ni agregan una característica.
- revert: Reversión a un commit previo.
- style: Correcciones de formato, comas, puntos, etc.
- test: Agregación de pruebas.

Área donde Afecta

Esta parte indica a los desarrolladores dónde surtió efecto el cambio. Por ejemplo, si el cambio fue en una parte gráfica, se debe tener una estructura como feat(graphics):. Si el cambio fue en el login, sería fix(login):. El área donde afecta debe ir en inglés, entre paréntesis y precedido por un signo de dólar \$ (aunque este símbolo no tiene una función específica, se ha adoptado como buena práctica).

Mensaje/Descripción del Cambio

El mensaje del commit debe tener una longitud máxima de 72 caracteres, describiendo claramente el cambio, corrección o agregación realizada. Debe estar en español para conveniencia del grupo y no debe iniciar con letras mayúsculas.

IDE a Utilizar (Sugerencia)

Para el desarrollo de web apps, el mejor IDE que se puede utilizar es Visual Studio Code. Este ofrece mucha versatilidad en cuanto a extensiones y manejo de lenguajes. A continuación, se exponen algunas razones por las cuales es conveniente:

Ventajas

- Extensibilidad
- Interfaz amigable
- Soporte multiplataforma
- Integración con Git
- Depuración avanzada

Desventajas

- Consumo de recursos
- Curva de aprendizaje
- Soporte limitado para proyectos grandes
- Limitaciones en comparación con IDEs completos

Aunque VS Code tiene ventajas y desventajas, es muy útil para nuestro proyecto. Aquí algunas extensiones recomendadas que facilitarán el desarrollo de Web-Apps:

- Code Runner: Provee soporte para una amplia gama de lenguajes, facilitando el autocompletado de funciones en HTML, CSS, JavaScript, Node.js, etc.
- Indent-Rainbow: Colorea las indentaciones para una correcta tabulación y jerarquización de las funciones.
- Live Server: Provee un servidor local para la visualización en tiempo real de los cambios en el código (se actualiza la página cada vez que se guarda).