



Listado resumen de las convenciones git

Estudiante: Eduardo Krause

+Convenciones git:

Git es una herramienta de control de versiones potente que permite a varias personas trabajar en un proyecto, y por eso mismo, es importante que se especifique de manera clara cada acción que sucede, para eso se definió una serie de reglas que se deben seguir a la hora de interactuar con el entorno de git al momento de realizar un commit, a continuación están las principales:

-Formato de los mensajes:

a la hora de realizar un commit se utiliza el siguiente formato de mensaje:

<type>(<scope>): <subject>

<línea en blanco>

<body>

<línea en blanco>

<footer>

-type: indica el tipo de campo realizado, para esto existen algunas etiquetas

- feat: indica adición de una funcionalidad

- fix: indica corrección de bug

- docs: indica adición de documentación

- style: indica corrección en la indentación

- refactor: indica refactorización de una lógica/función

- test: indica adición de un test faltante

- chore: indica mantenimiento del código

-scope: indica que parte de proyecto fue modificada

-subject: indica que cambio se realizó, se recomienda no pasar de los 100 caracteres y utilizar un lenguaje en modo imperativo

-Body: descripción detallada del cambio realizado, cada línea no debe superar 72 caracteres, debe responder las preguntas ¿qué cambio? y ¿por qué?

-footer: indica los cambios críticos y los que pueden romper cosas dentro del proyecto, sigue las mismas reglas que el body, con la adición de que debe preceder la frase BREAKING CHANGE

Ejemplos de commit:

-ejemplo1:

Add funcion para registro de datos

Add (Main): se implementa una función que permita registrar los datos de la aplicación

se crea una función nueva dentro del código que solicita valores desde otras funciones y los registra para usos futuros

-ejemplo2:

Test validacion de numeros

Test (branch 3): agregar un test que permita validar los números dados

se crea un test que se asegura que cada valor esté dentro de los límites estipulados

BREAKING CHANGE: la aplicación se caerá en caso de fallar en recuperar los valores.

-ejemplo3:

Drop validacion de numeros

Drop (branch 3): se elimina el test que valida los números

Se elimina el test que valida los números, ya que posteriormente se implementará en función.

-ejemplo4:

fix variables de números

fix (Main): se corrigió el tipo de variable que guarda los números para que permita decimales

El tipo de variable que guarda los datos numéricos se cambió de int a float para permitir un mayor rango de aplicaciones.

BREAKING CHANGE: la aplicación se cae en caso de recibir un dato no numérico.

-ejemplo5:

Feat función de suma

feat (branch 4): se agrega la funcionalidad que permite sumar los valores de las variables

se agrega una función que obtiene valores desde otras funciones, y se encarga de sumar estas variables

BREAKING CHANGE: salta un error en caso de tratar de operar varios tipos de variables.

+Bibliografía:

-<https://github.com/KarmaPulse/git-commit-message-conventions/blob/master/README.md>