Programar es difícil, pero todos podemos hacerlo

Frontend vs Backend

Frontend y **Backend** son términos que se refieren a la separación de intereses entre una capa de presentación y una capa de acceso a datos


FRONT END
BACK END
USER
comic.browserling.com

¿Qué tecnologías usamos en Frontend?

- HTML
- CSS
  - Preprocesadores de CSS (SASS, LESS, Stylus)
- Javascript
  - Librerías y Frameworks (jQuery, ReactJS, AngularJS, etc)

Ejemplos

Envío Gratis con tu compra superior a $2.000!

- **SAN VALENTIN!**

- Inicio
- Productos

- Marcas

ENVÍO GRATIS CON TU COMPRA SUPERIOR A $2.000!

TAN Intensa

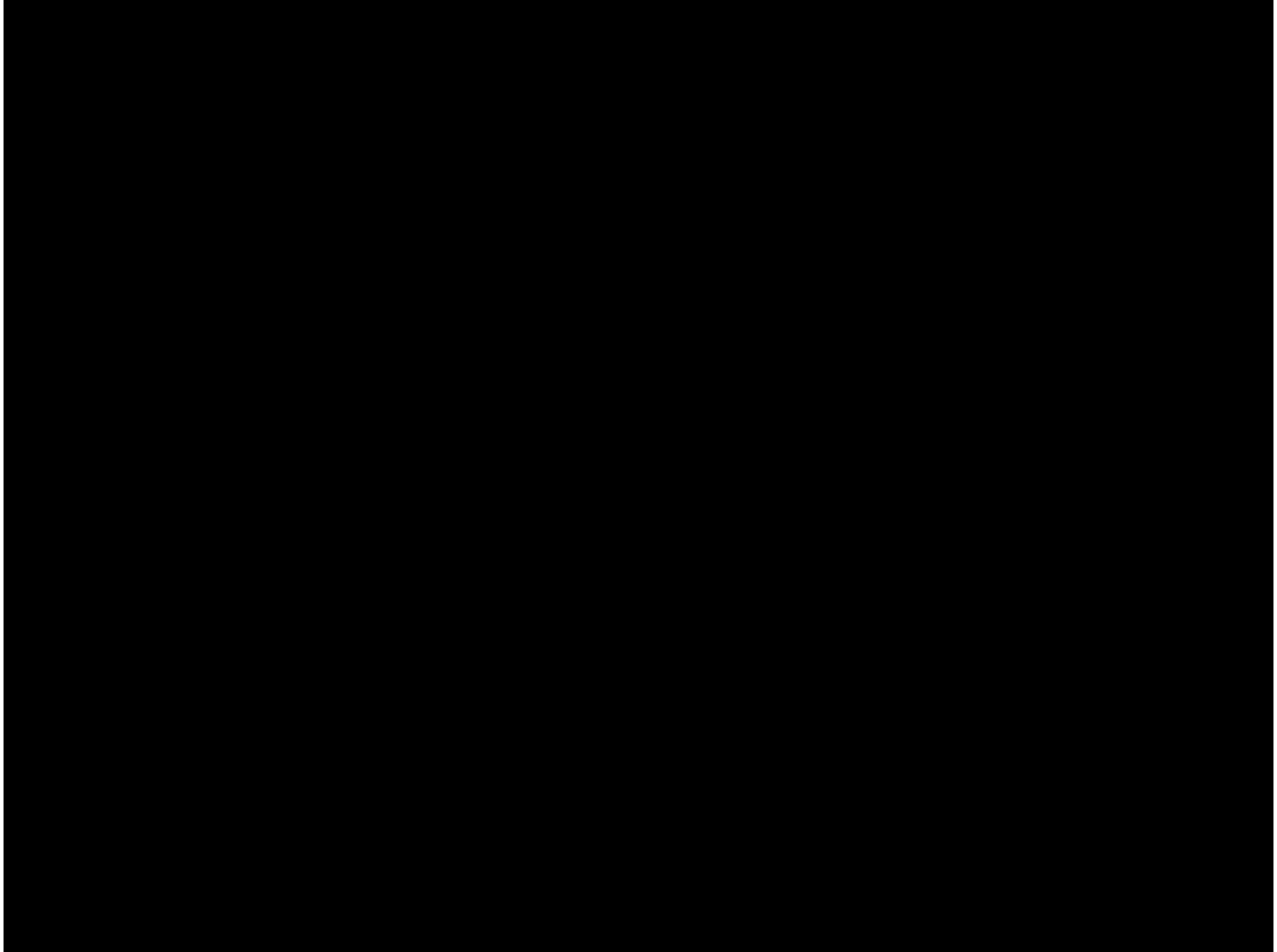
SAN VALENTIN! INICIO PRODUCTOS MARCAS

Encontra tu prenda ideal

PAGO EN EFECTIVO O CON TARJETA
VER MEDIOS DE PAGO DISPONIBLES

ENVIOS A TODO EL PAIS
CALCULAR COSTO DE ENVIO

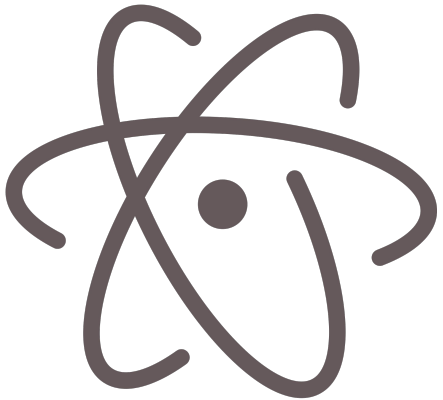
RETIRO EN SUCURSAL
VER DIRECCION



Herramientas de trabajo

Editores de texto / IDE

Sublime Text 3

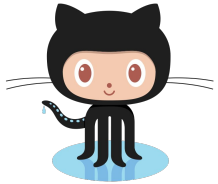


Atom

Visual Studio Code

Control de versiones

El control de versiones es un sistema que registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo. El principal beneficio de versionar los archivos es la capacidad de que dos usuarios puedan trabajar sobre un mismo archivo al mismo tiempo



www.github.com

Terminología

Repositorio

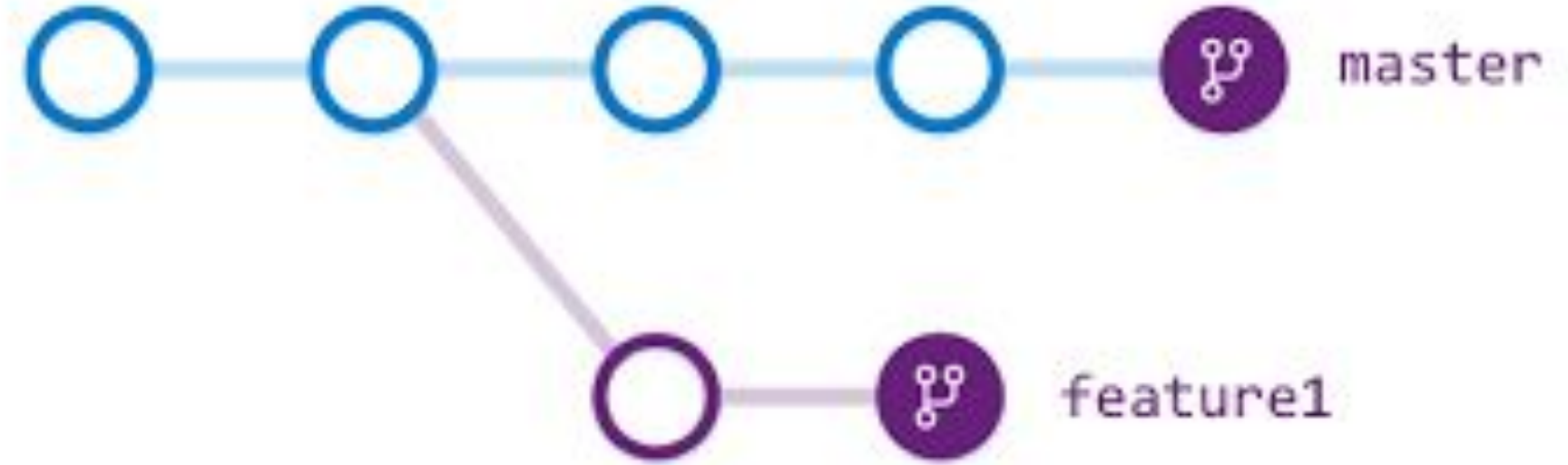
El **repositorio** es el lugar en el que se almacenan los datos actualizados e históricos de cambios, a menudo en un servidor.

Versión

Una **versión** es un estado determinado de el archivo que estamos revisando. Las versiones se identifican mediante un código de detección de modificaciones (Ej. Git usa SHA1). A la última versión se le suele identificar de forma especial con el nombre de **HEAD**.

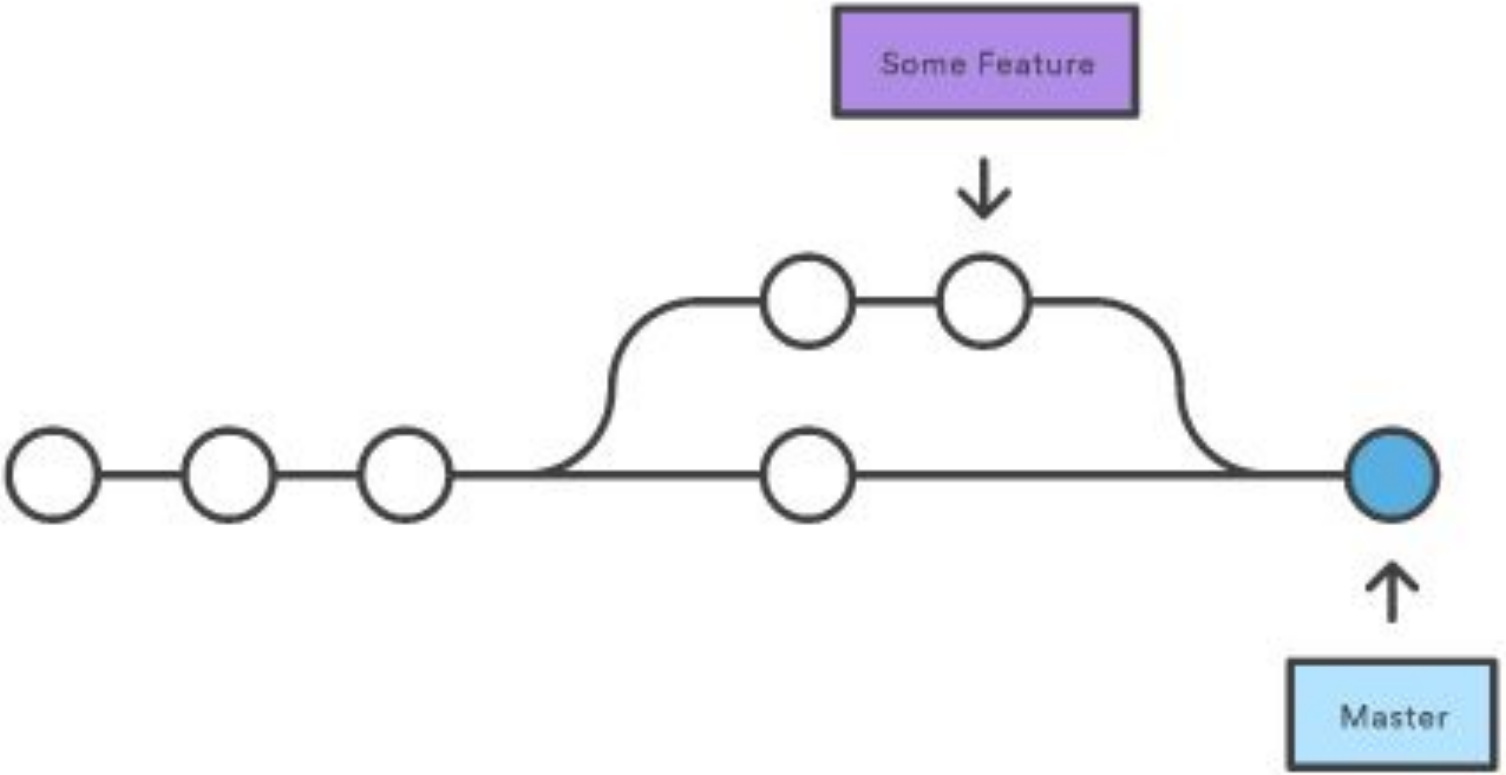
Branch

Es una bifurcación de una rama dentro de un repositorio. Todos los cambios realizados sobre la bifurcación no se aplican en la rama principal sino hasta realizar un *merge*

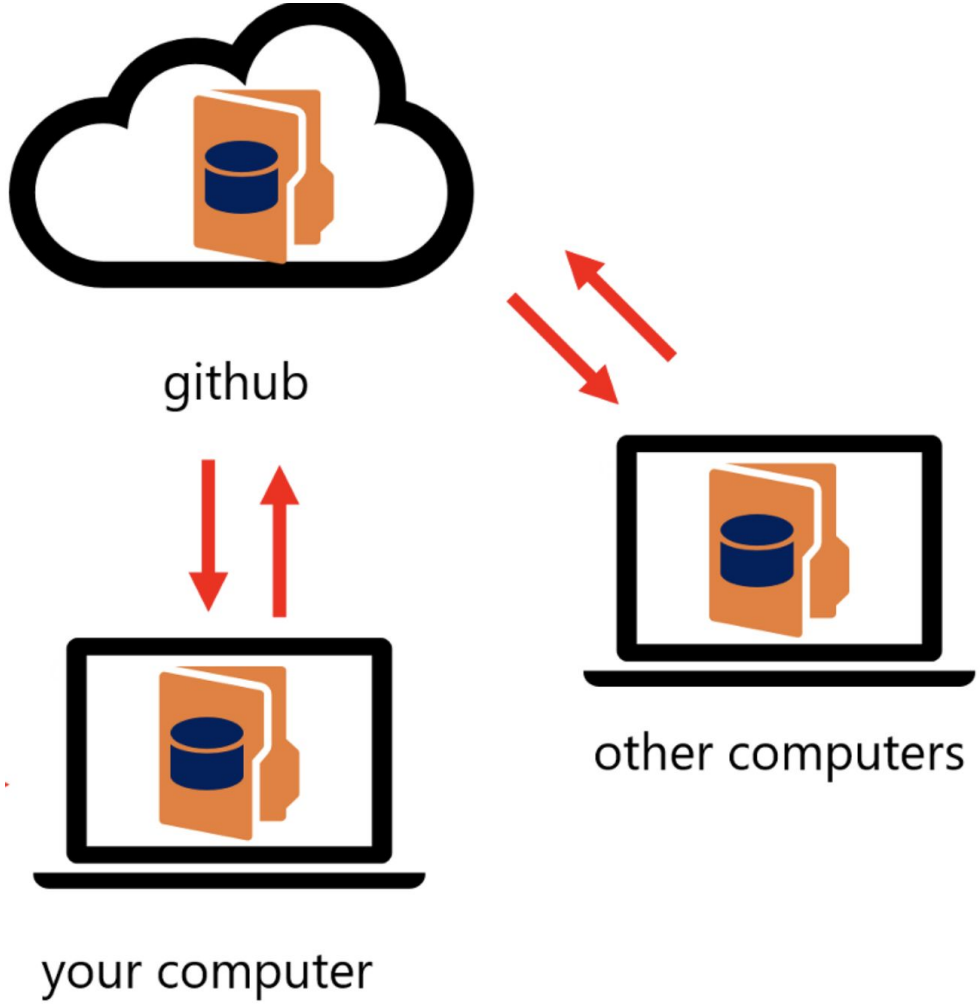

master
feature1

Merge

Es la unión entre dos ramas realizada desde una de las mismas.


Some Feature
Master

¿Cómo funciona GIT?


github
other computers
your computer

Comandos básicos

GIT status: me dice si tengo cambios para subir.

GIT add: agrega mis cambios para que pueda luego
subirlos. (Si hay archivos nuevos, modificados)

GIT commit: genera una versión con mi código fuente.

GIT push: empuja al repositorio remoto las versiones que tengo en mi repositorio local.

GIT pull: me trae los cambios del repositorio remoto.

Archivo1.html

Archivo2.html

Al hacer el **pull**, me traigo los cambios sobre el archivo2.html, como mis cambios locales fueron sobre archivo1.html, se actualiza automáticamente.

GIT pull: me trae los cambios del repositorio remoto.

2

Archivo1.html

Archivo1.html

Al hacer el **pull**, me traigo los cambios sobre el archivo1.html, mis cambios locales fueron sobre el mismo archivo, se deberá hacer un Merge.

Merge Automático

Git ordena automáticamente los cambios locales con los cambios de la copia traída del repositorio remoto.

Archivo1.html (cambio línea 38)

Archivo1.html (cambios líneas 56 y 60)

Si los cambios fueron realizados en distintas líneas del archivo, el proceso se resuelve automáticamente.

Merge Manual

Git no puede ordenar automáticamente los cambios locales con los cambios de la copia traída del repositorio remoto, dado que se ve afectada la misma línea.

Archivo1.html (cambio línea 38)

Archivo1.html (cambio línea 38)

GIT pedirá que se revisen los cambios del archivo manualmente y que luego se commitee el archivo resultante.