TSR - PRACTICA 1

Grupo:

L1-3A

Equipo:

Segarra Orenga, José María Cardo Sancho, Carlos Bustos Ferrer, José Vicente Abellán Romero, Francisco José

URL del dep. Canónico:

git@gitlab.dsic.upv.es:jobusfe1/ practica1_canonico.git

Tabla de Contenidos:

ProyectoBasico ProyectoEjemplo0 Pre-Proxy TCPproxy Proxy

PROYECTO BÁSICO

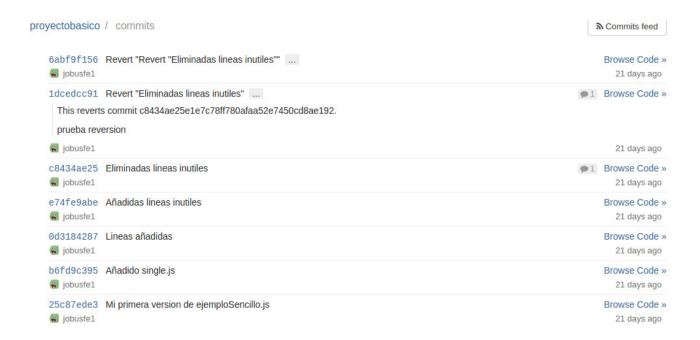
git@gitlab.dsic.upv.es:jobusfe1/proyectobasico.git

Descripción del proyecto

Probar Git mediante subida, actualización y recuperación de versiones anteriores de los archivos single.js y eventSimple.js

Objetivo

Conocer los comandos de Git y saber cuando emplearlos para el correcto funcionamiento. No hemos tenido problemas relevantes



PROYECTO EJEMPLO 0

git@gitlab.dsic.upv.es:jobusfe1/proyectoejemplo0.git

Descripción del proyecto

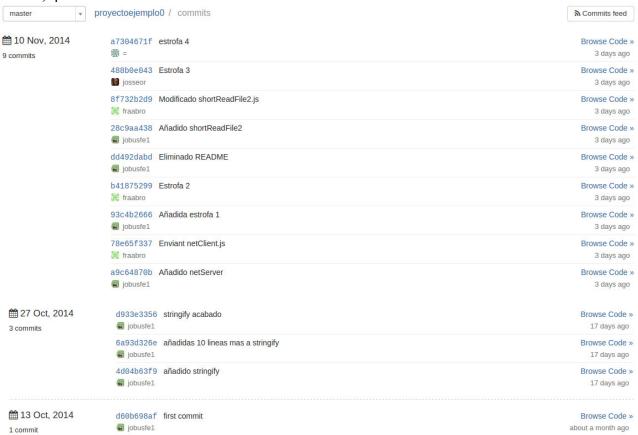
Edición progresiva y complementaria de archivos en Git.

Problemas relevantes y su solución

Problemas de conflictos entre archivos subidos por diferentes desarrolladores.

Solución: coger buenos hábitos en Git.

- 1.- Comunicar cambios al resto del equipo.
- 2.- Reinicializar el depósito local ("git init").
- 3.- Comprobar que está configurado el depósito remoto con "git remote -v". Si no, añadirlo mediante "git remote add [nombre] [URL_deposito]"
- 4.- y actualizarlo con la copia del remoto: "git pull [nombre] master"
- 5.- Cada desarrollador modifica su propio archivo, sin interferir con los de los demás, y lo añade/ actualiza a su depósito local mediante "git add [fichero]"
- 6.- Se realiza un commit con los cambios realizados.
- 7.- Se sube al depósito remoto grupal ("git push -u [nombre] master") para su revisión por el master, que actualiza su local con los nuevos cambios.



PRE PROXY

git@gitlab.dsic.upv.es:jobusfe1/preproxy.git

El proyecto de pre-proxy consite en crear un cliente y un servidor. El cliente se conecta al servidor que tiene configurado enviando un mensaje con su dirección IP. El servidor recibe dicha petición y le devuelve al cliente la IP Cliente, la IP Servidor y el numero de carga del trabajo actual.

Problemas relevantes y su solución:

Los problemas encontrados fue la utilización del JSON. La solución al problema fue la mera documentación extra en el API.

Lineas Relevantes:

CLIENTE:

SERVIDOR:

TCPPROXY

git@gitlab.dsic.upv.es:jobusfe1/tcpproxy.git

El proyecto de tcp-proxy consiste en crear un intermediario (proxy) entre un cliente y un servidor. El cliente en vez de conectarse al servidor directamente (como en el pre-proxy) se conecta al proxy en un puerto especifico (8001) y el proxy redirige a la IP y puerto que haya sido configurado previamente mediante el programa "programador.js"

Problemas relevantes y su solución:

El problema relevante en dicho proyecto fue como configurar en caliente la regla del proxy ya que cuando se usaba el programador.js se procedía a configurar la regla y directamente la ignoraba o saltaba Error. La solución fue la creación del nuevo Socket como se muestra a continuación.

Lineas Relevantes:

PROXY

git@gitlab.dsic.upv.es:jobusfe1/proxy.git

Dicho proyecto consiste en mejorar el proxy realizado anteriormente (proyecto TCP Proxy) para que se puedan configurar múltiples reglas usando diferentes puertos.

Problemas relevantes y su solución

El problema relevante en este proyecto fue el planteamiento para poder hacer varias reglas. La solución fue la mera experiencia en otras asignaturas.

Lineas relevantes

PROXY.JS:

CONTROLADOR:

```
var client = net.connect( port, IP_PROXY, function () {
          var envio = JSON.stringify ({'op':'set', 'inPort':PORT_PROXY, 'remote': {'ip':IP_REMOTE,
          'port': PORT_REMOTE}});
          client.write(envio);
          client.end();
});
```