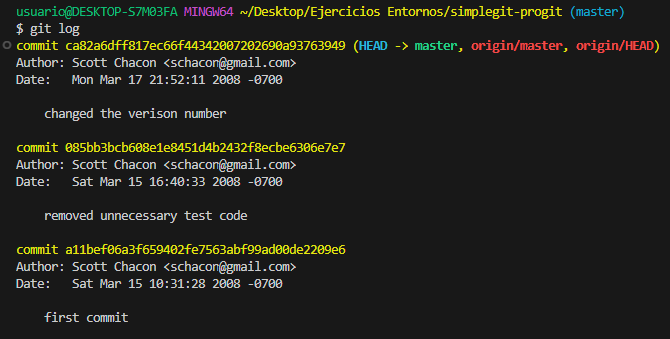
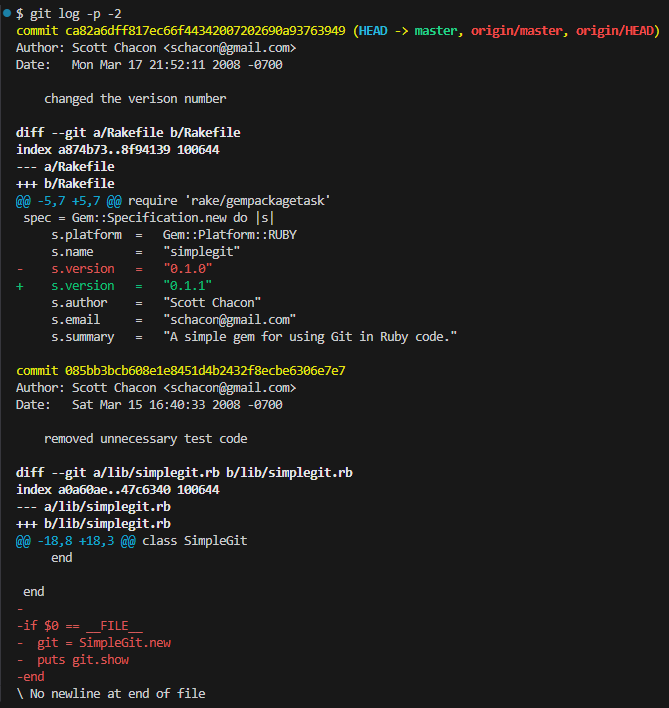
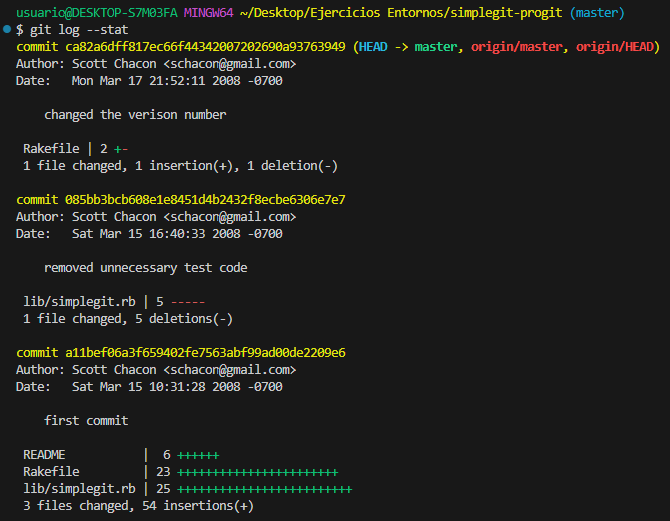
El primer comando usado a parte del clone es "git log" que muestra el historial de cambios que ha tenido el repertorio en orden cronológico inverso



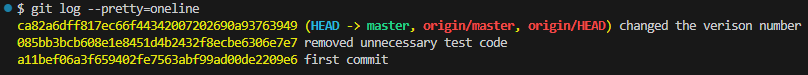
El siguiente comando mencionado en el ejercicio es "git log -p -2" la parte "-p" muestra las diferencias introducidas en cada cambio,y la parte de "-2" indica que solo mostrara las dos últimas entradas

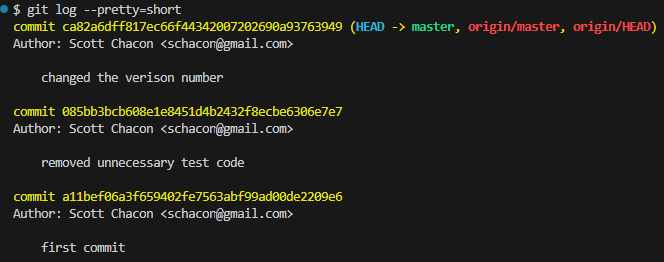


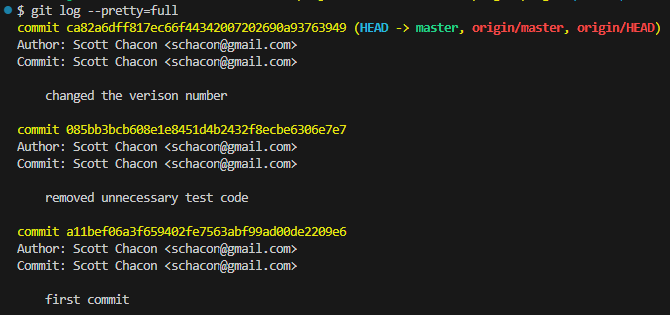
El siguiente comando es "git log --stat" que muestra tras cada confirmación los archivos modificados y cuantas lineas se han añadido y eliminado y donde ha sido

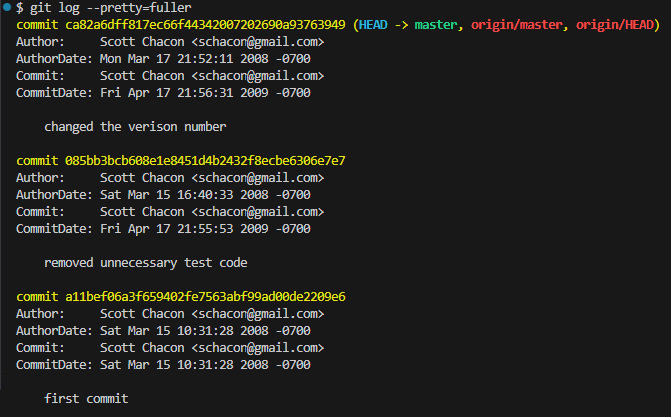


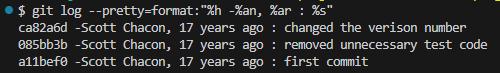
El comando "git log --pretty" modifica el formato en la que se muestra la información dentro de este comando puedes elegir entre:

"git log --pretty=oneline" que mostrará cada confirmación en una única línea, pero no enseña los cambios que se han efectuado.

"git log --pretty=short" que muestra un poco más de información que "oneline" siendo esta el nombre y email del autor del repertorio.

"git log --pretty=full" que muestra tanto el nombre y email del autor como del editor que ha realizado el commit.

"git log --pretty=fuller" aparte de la información mostrada en el "full" también muestra la fecha y la hora en la que estas acciones se han efectuado.

El siguiente comando mencionado es otro apartado del "--pretty", git log --pretty=format", que permite especificar tu propio formato.

Para realizar este comando tienes que especificar de que forma quieres que te muestre los datos. Algunos ejemplos que muestra el ejercicio son:

%H Hash de la confirmación

%h Hash de la confirmación abreviado

%T Hash del árbol

%t Hash del árbol abreviado

%P Hashes de las confirmaciones padre

%p Hashes de las confirmaciones padre abreviados

%an Nombre del autor

%ae Dirección de correo del autor

%ad Fecha de autoría (el formato respeta la opción -–date)

%ar Fecha de autoría, relativa

%cn Nombre del confirmador

%ce Dirección de correo del confirmador

%cd Fecha de confirmación

%cr Fecha de confirmación, relativa

%s Asunto

Estas opciones se introducen entre comillas dobles ej:

git log --pretty=format:"%h"

git log --pretty=format:"%h - %an, %ar : %s"

El comando "graph" se utiliza especialmente con "oneline" y "format", y añade un pequeño gráfico ASCII mostrando el historial de ramificaciones y uniones.

"git log" tiene muchas más opciones, algunas que no se muestran especialmente en el ejercicio por ahora son:

"-p Muestra el parche introducido en cada confirmación.

"--shortstat" Muestra solamente la línea de resumen de la opción --stat.

"--name-only" Muestra la lista de archivos afectados.

"--name-status" Muestra la lista de archivos afectados, indicando además si fueron añadidos, modificados o eliminados.

"--abbrev-commit" Muestra solamente los primeros caracteres de la suma SHA-1, en vez de los 40 caracteres de que se compone.

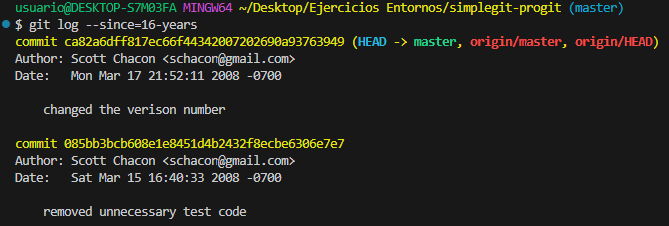
"--relative-date" Muestra la fecha en formato relativo (por ejemplo, “2 weeks ago” (“hace 2 semanas”)) en lugar del formato completo.

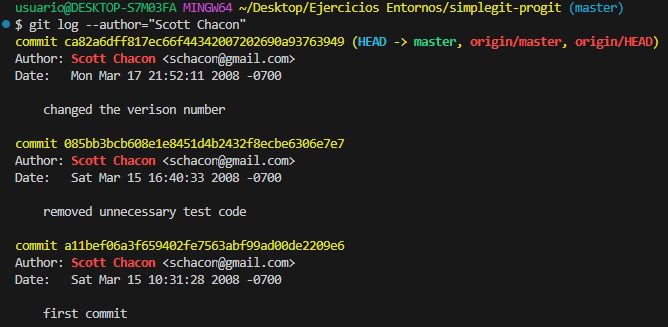
"git log" acepta opciones para limitar la cantidad de confirmaciones que queremos que muestre, siendo el anteriormente usado "-2" una de ellas.

También te puede mostrar una cantidad de confirmaciones definida en el tiempo, siendo "--since" y "--until" dos de ellas.

"--since" se usa para especificar desde qué punto queremos partir.

"--until" se usa para especificar hasta qué punto queremos partir.



También se puede filtrar para que muestre solo las confirmaciones que cumplan ciertos requisitos, como “--author” que filtra por autor, “--commiter” que filtra por usuario que ha realizado el commit, “--greg” que filtra por palabras clave entre los mensajes de confirmación.

Para utilizar varias de este estilo a la vez, hay que tener en cuenta el comando “--all-match” para que muestre solo las que coincidan en todos casos que especifiquemos y no muestre las que solo coincidan en uno de ellos

Otra opción es “-S” la cual recibe una cadena y muestra las confirmaciones que cambiaron el código añadiendo o eliminando la cadena.

La ultima opción que nos da el ejercicio es buscar directamente en la dirección de la cual queremos ver los cambios