



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

ALEJANDRO PIMENTEL

*Profesor:*

FUNDAMENTOS DE PROGRAMACIÓN

*Asignatura:*

BLOQUE 135

*Grupo:*

3

PRACTICA 1

*No de Práctica(s):*

ALITZEL ANAID GUTIÉRREZ RAMOS

*Integrante(s):*

*No. de Equipo de  
cómputo empleado:*

9370

*No. de Lista o Brigada:*

1er SEMESTRE

*Semestre:*

26-08-2019

*Fecha de entrega:*

*Observaciones:*

Excelente, te recomiendo que pongas mas descriptiones en la parte del desarrollo junto a las imágenes para que no olvides el significado de cada instrucción.  
Además, no olvides incluir una sección de conclusiones e introducción, todo trabajo escrito los necesita

**CALIFICACIÓN:** 10

## OBJETIVOS

Aprender, entender y practicar las diferentes herramientas que nos brinda software para un mejor desempeño tanto en nuestra vida estudiantil como futura profesional ya qué dando una introducción; esto nos sirve tanto para crear, mandar y modificar archivos en la nube donde tenemos cualquier tipo de servicio y almacenamiento accesibles mediante internet ya que son redes de servidores interconectado para operar de forma segura e interrumpida, como vendría siendo en la plataforma de GitHub.

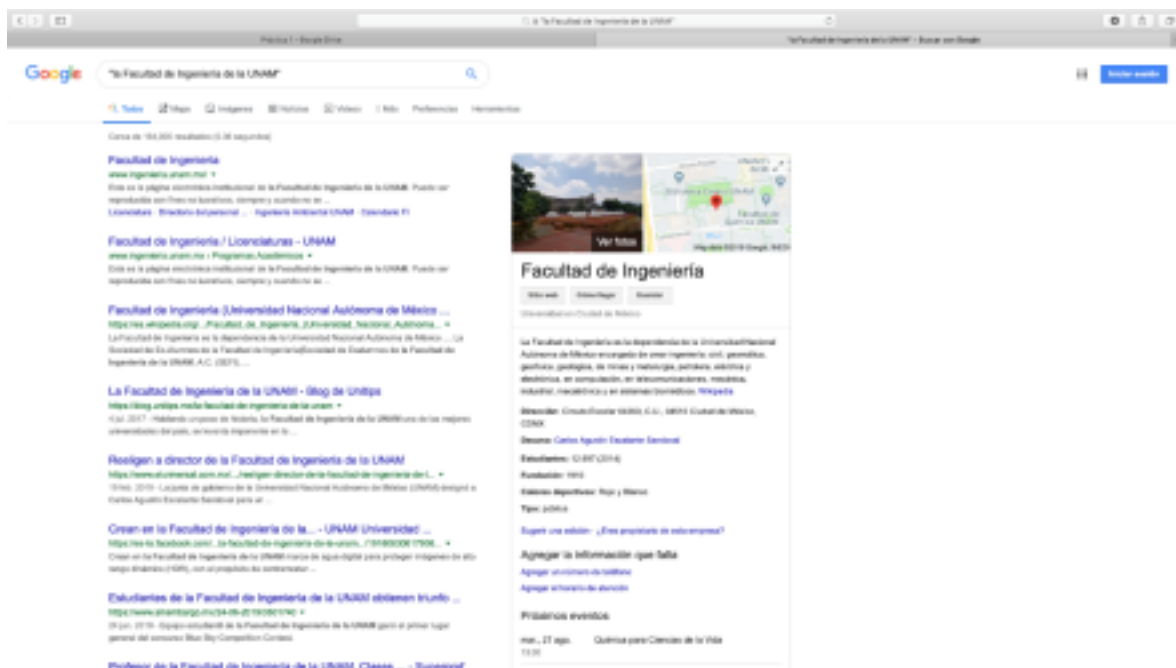
Tanto así, como realizar investigaciones más concretas y específicas con comandos avanzados en la plataforma de Google y finalmente saber crear cualquier tipo de trabajos, ensayos, investigaciones; como una tesis, de una forma más organizada con el uso de Control de versiones con Git.

## ACTIVIDADES DESARROLLADAS

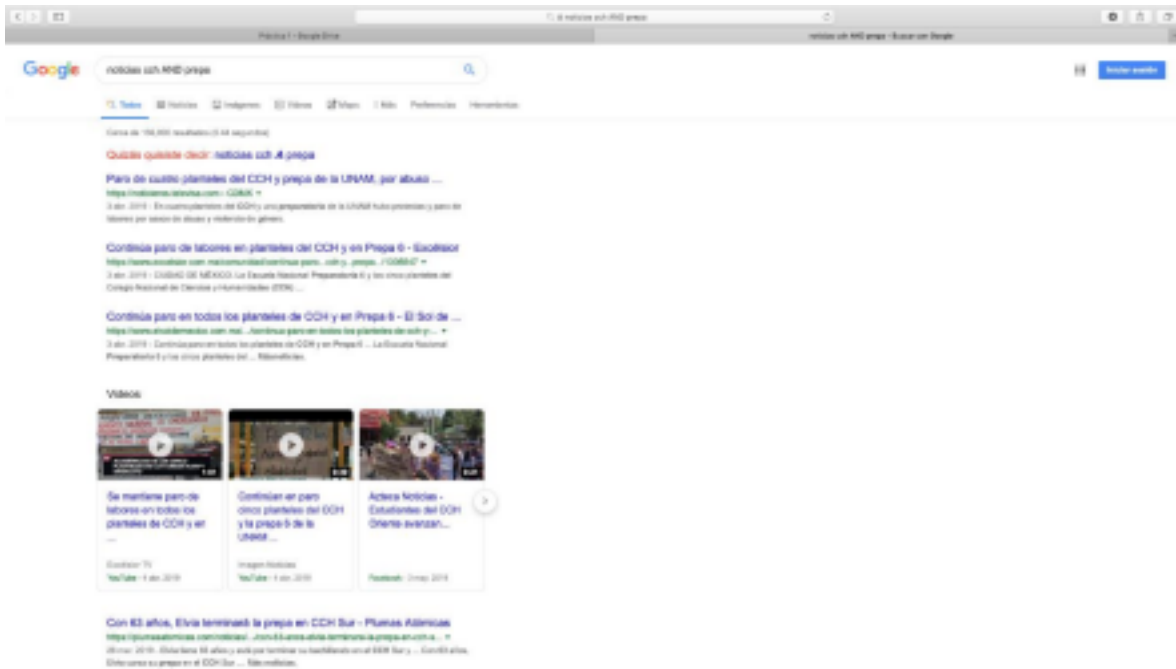
Empezamos con la búsqueda y captura de diferentes tipos de ejemplos con comandos avanzados.

1. Primero entramos a nuestra plataforma Google en nuestra PC.
2. Capturamos las siguientes combinaciones

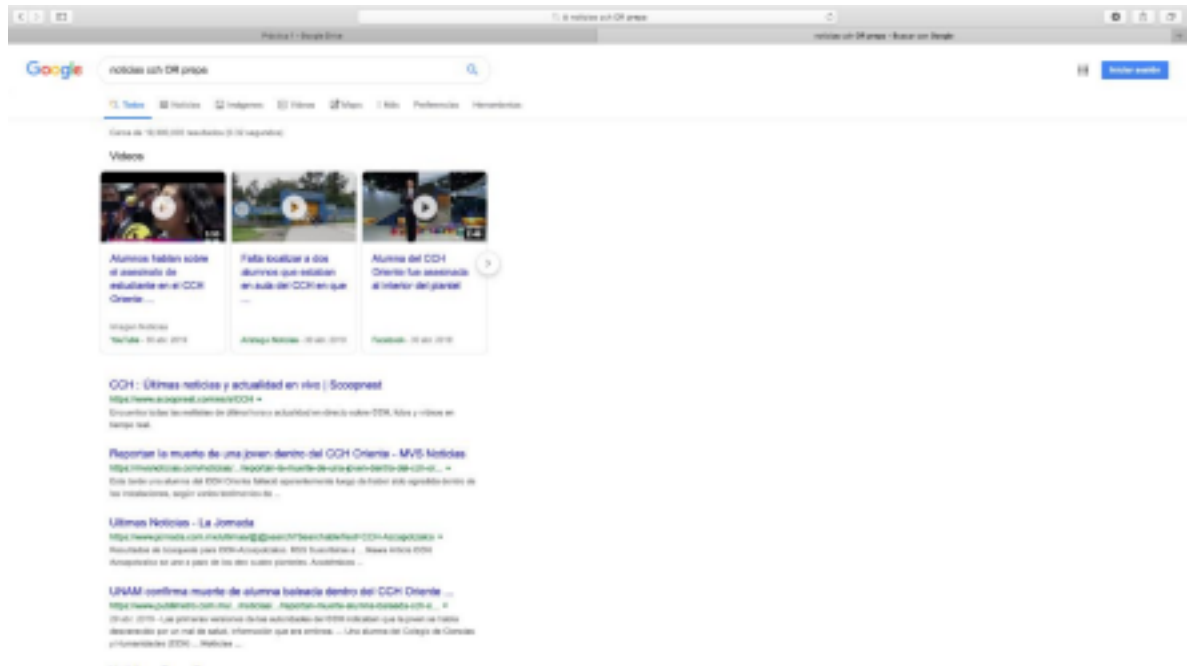
- ""



- AND

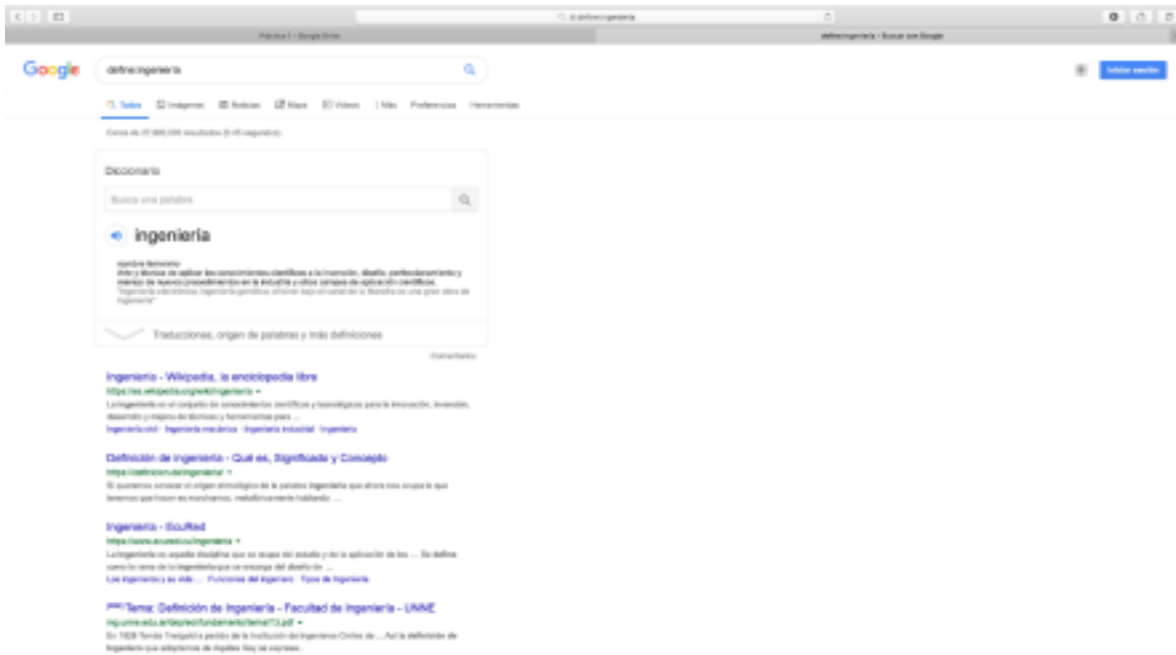


- OR

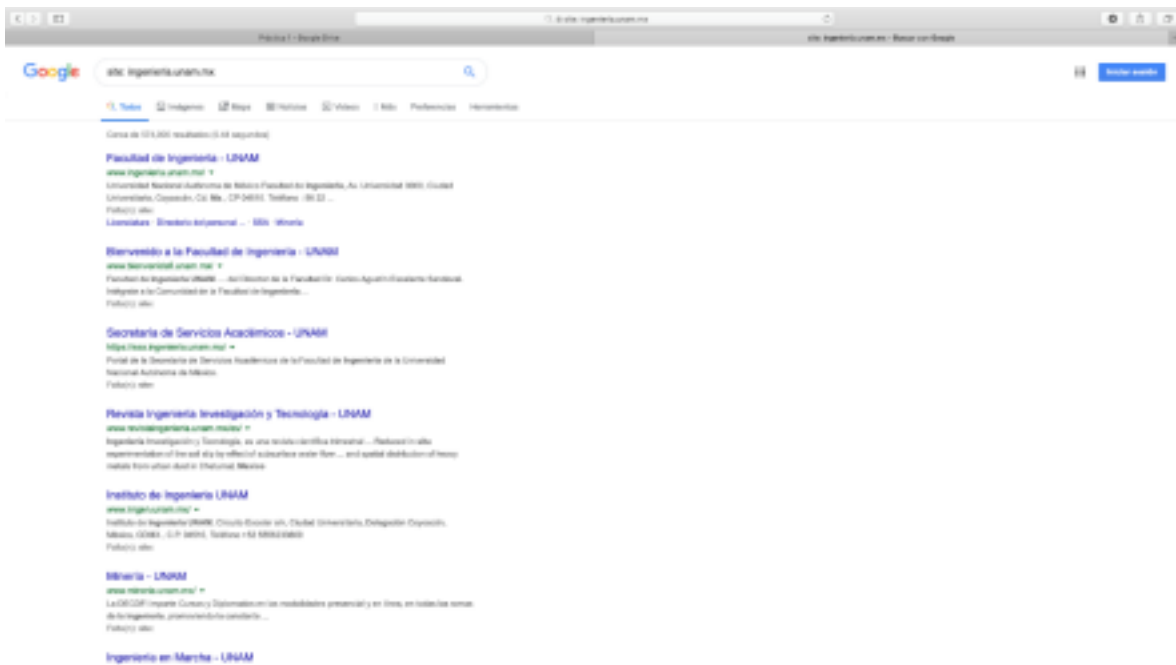




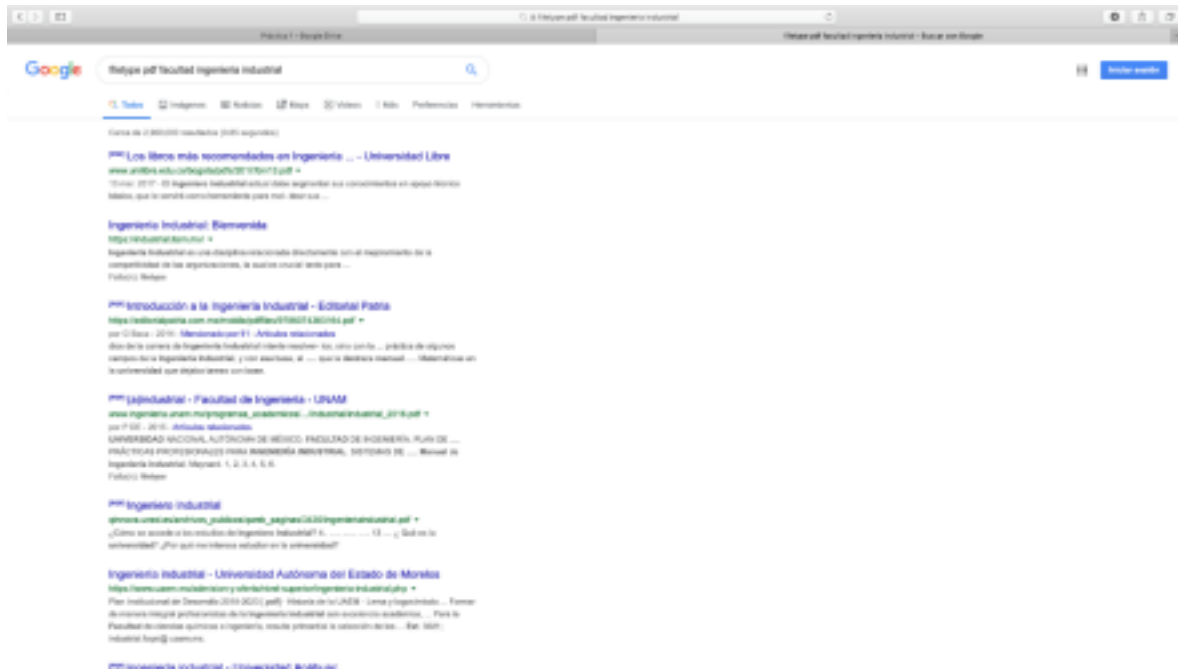
- define:



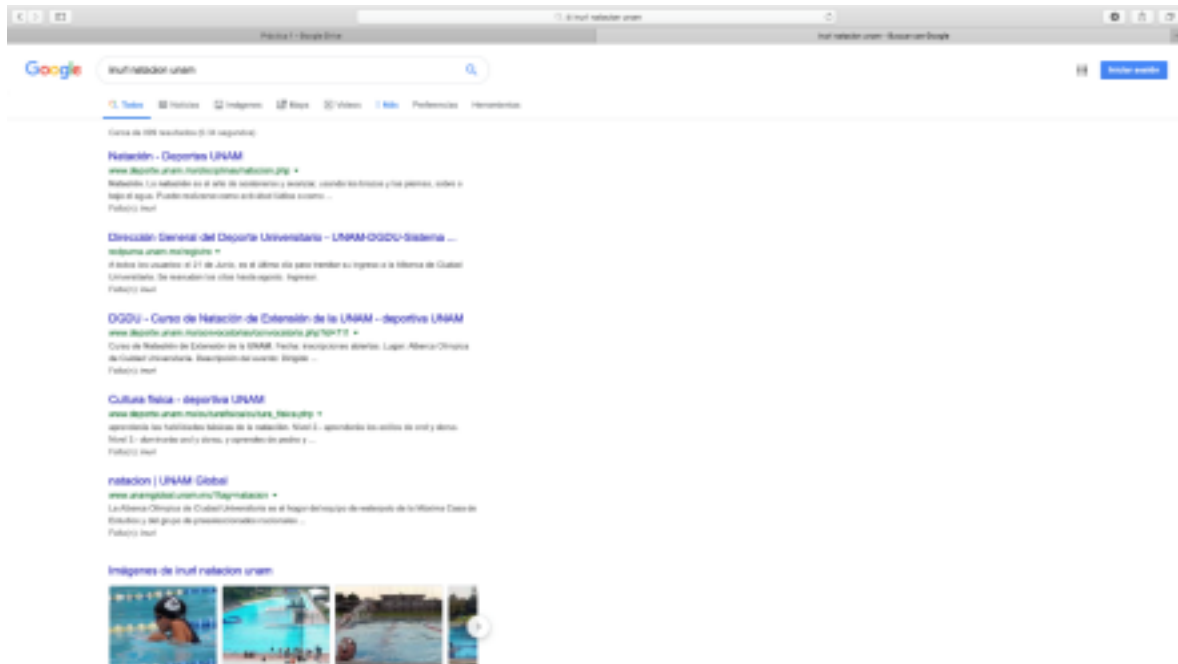
- site:



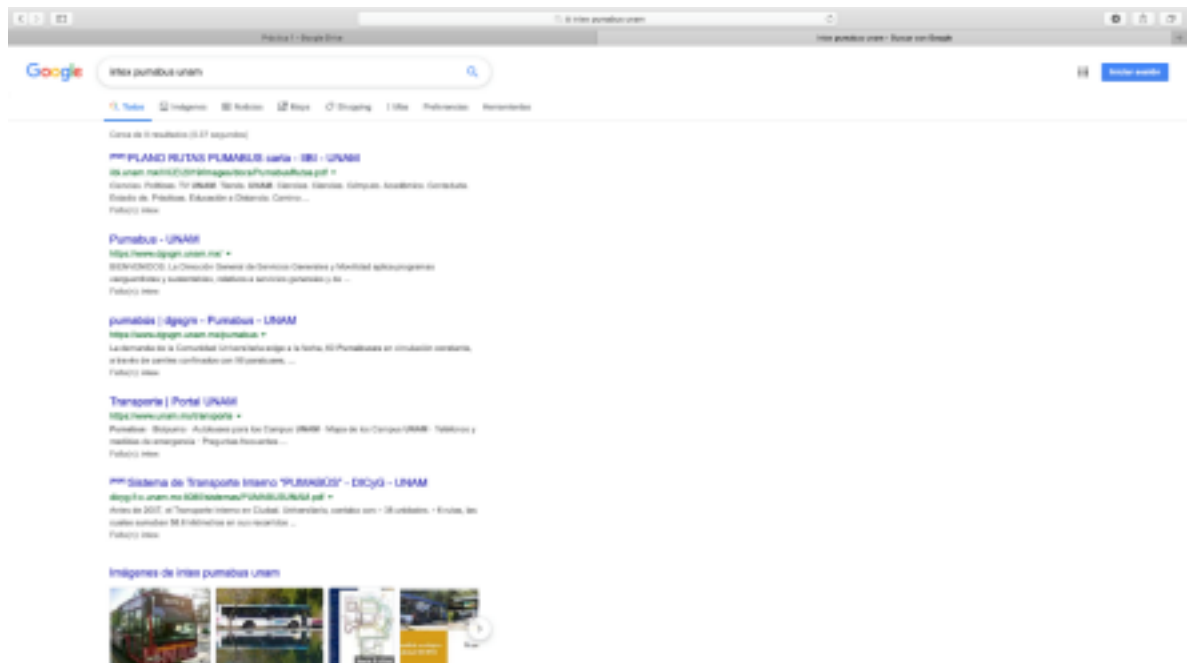
- filetype:



- inurl:



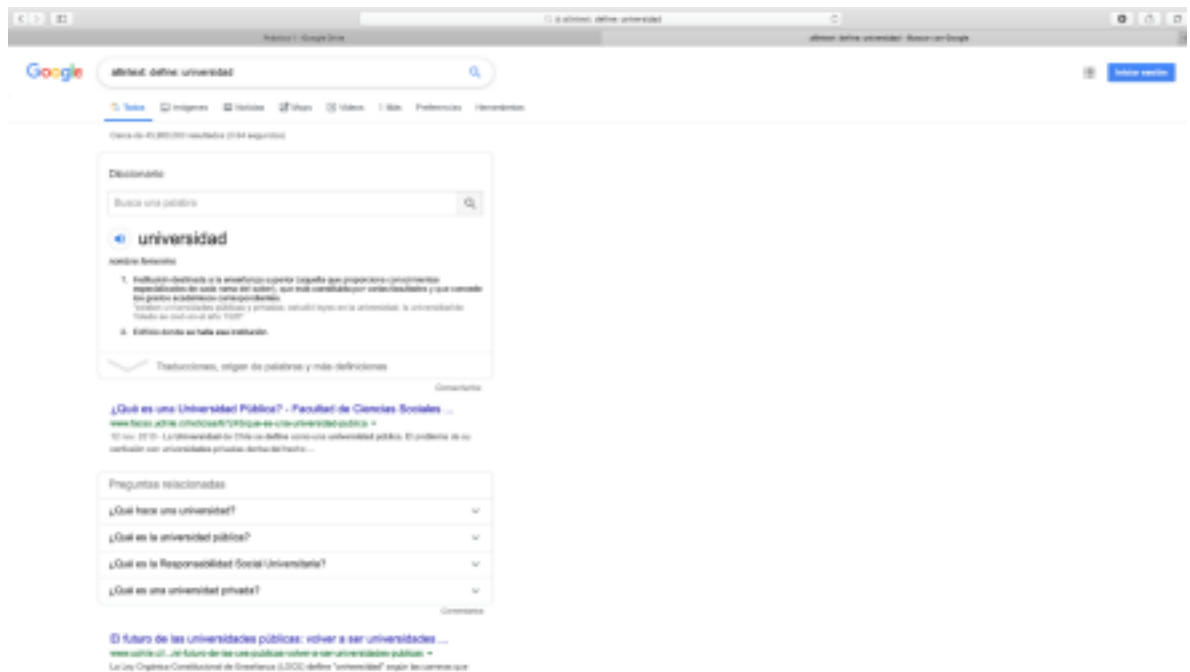
- intext:



- AROUND(2)



- Mis 3 diferentes combinaciones:



i.

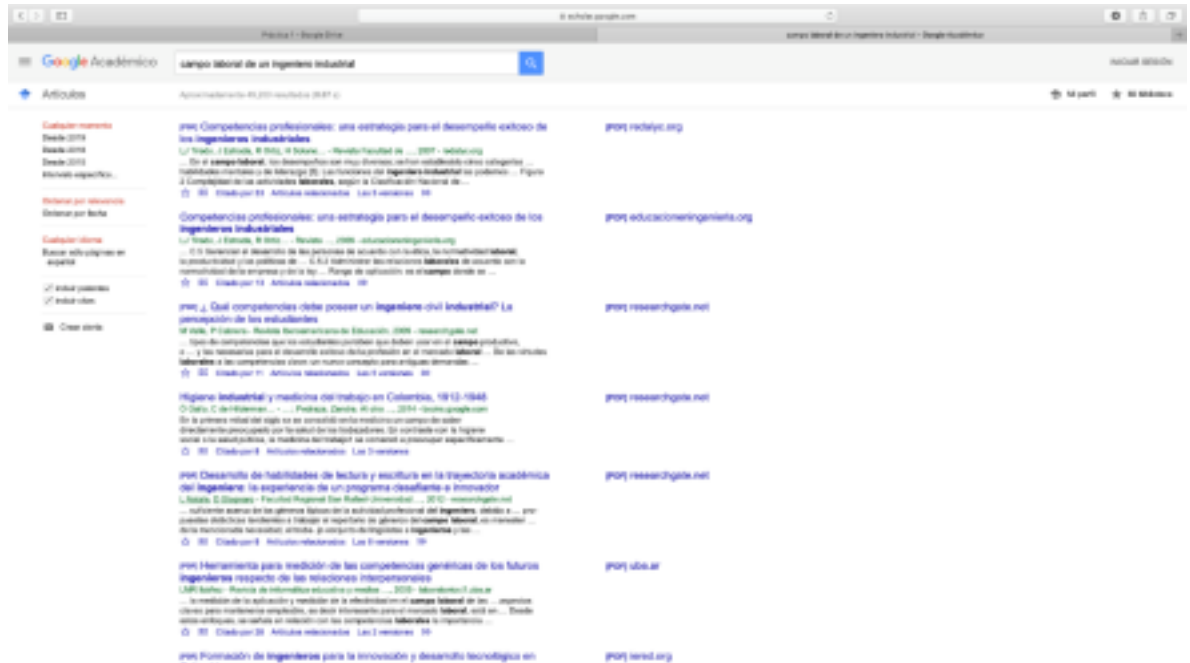




**GOOGLE SCHOLAR**

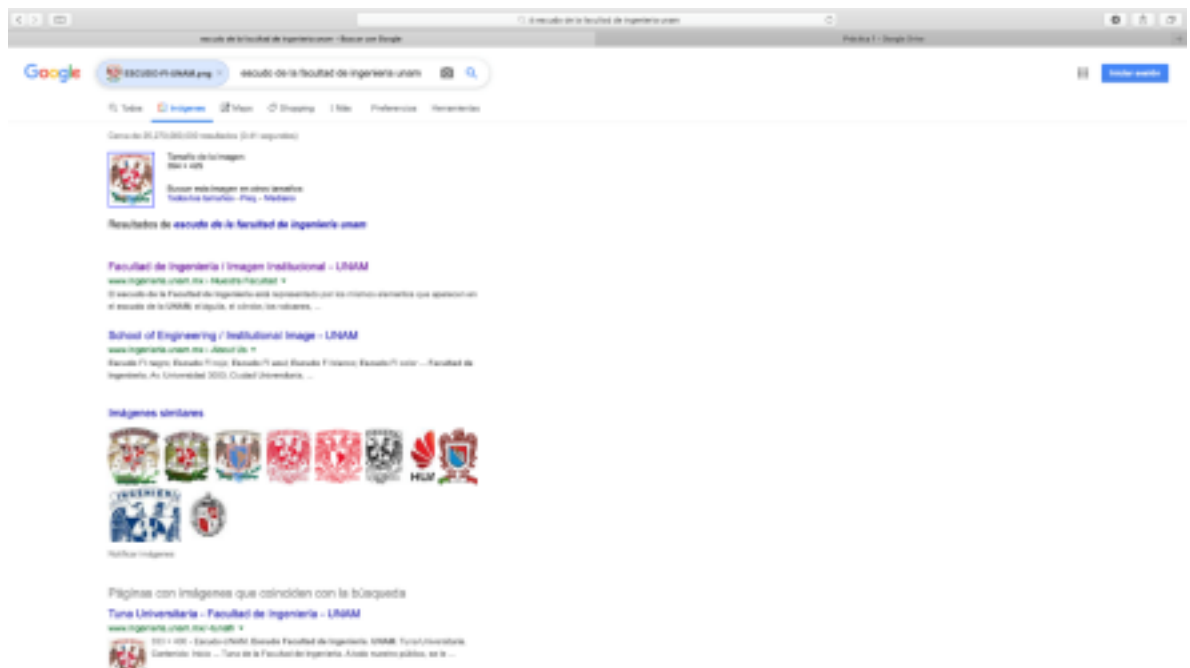






v.

Buscando así, en el mismo Google Académico dos imágenes; una de mi facultad y otra de mi agrado.



i.



4. Finalmente en el laboratorio nos dimos a la tarea de conocer y poner en práctica algunos controles de versiones.

- Empezamos con una configuración inicial utilizando git, ya utilizamos una entrada de texto donde nos identificamos y registramos cambios.

Ejemplo:

`$ git config --global user.name` ➡ identifique al programador.

- Después iniciamos nuestro proyecto por así decirlo, donde se almacenó todo nuestro seguimiento:

`$ mkdir ~/PROYECTO`  
`$ cd ~/PROYECTO.`  
`$ git init`

} cree una carpeta oculta.

`$ git status` ➡ revise el estado de mis archivos

- Empecé a utilizar la instrucción add para que git tomara en cuenta los nuevos cambios que estaba por realizar:

`$ git add README` ➡ modifique el archivo para así ver el seguimiento

- Se utiliza la instrucción touch para añadir:

`$ touch archivo1,2,3,4,5...` ➡ añadí archivos y carpetas completas

- Es posible utilizar en especial .gitignore para dar seguimiento a algunos archivos en particular:

`$ echo "*.temp" > .gitignore` ➡ di seguimiento a todo excepto a algunos archivo

"\*" hace coincidir con cualquier cosa.

!" Para negar e ignorar todo lo que coincida en esa línea.

Tanto se puede iniciar como terminar una línea con el símbolo "/". Especifica que se trata de un directorio completo que se debe ignorar o para ignorar archivos en la carpeta raíz del proyecto y no haya recursividad.

- La forma en la que queda guardado el estado actual como una versión es utilizando el commit, es bueno utilizarlo muchas veces ya que este exige un mensaje:

\$ **git commit -m** " " ➡ narré todos los cambios registrados

- Dejando el git a un lado, solo se pone directamente:

\$ **rm** archivo ( )  
\$ **mv** archivo ( ) } borré y moví archivos

- Para sacar un archivo y no borrar la carpeta utilizamos cached y la instrucción HEAD para si quitar todos los archivos:

\$ **git rm --cached** (x) ➡ Saque el archivo x del seguimiento y no borre la carpeta.  
Gracias a cached

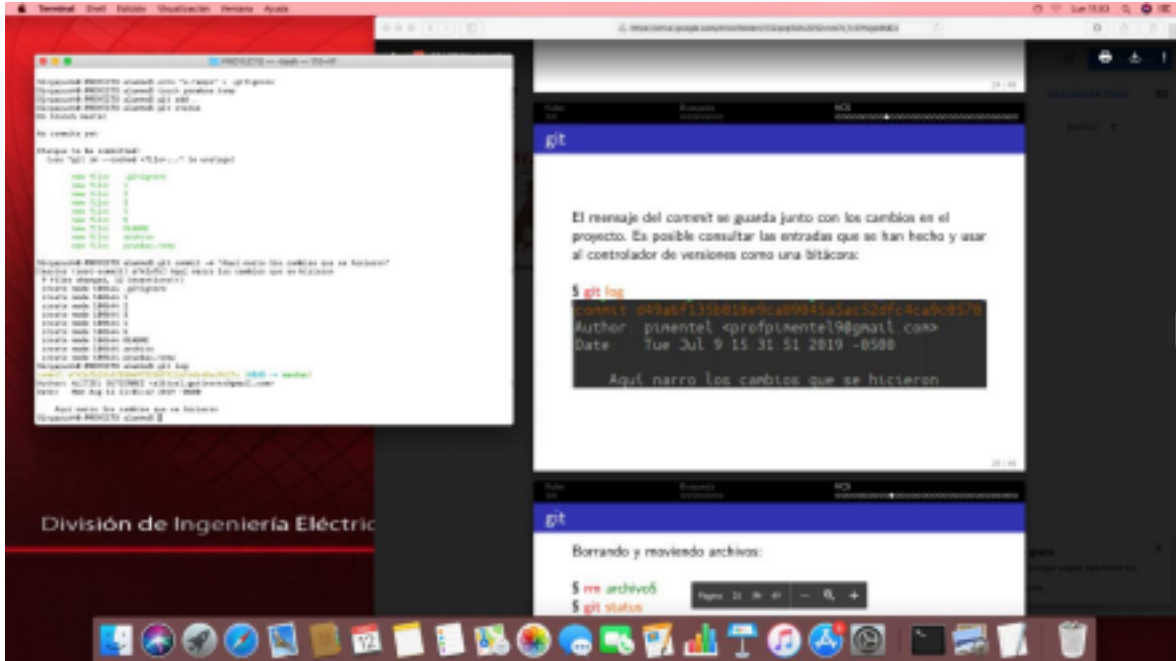
\$ **git reset HEAD** ➡ quité todos los archivos al mismo tiempo

\$ **git log** ➡ regresé a archivos

- Y finalmente utilizamos la instrucción checkout con argumento que indique al archivo que quieren revertir, no se debe dejar ningún espacio en blanco ya que ocasiona comportamientos inesperados:

\$ **git checkout** ➡ revertí un archivo

- Evidencia final:





5. La plataforma GitHub es para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de computadora. Donde realizamos una cuenta personal gratuita para empezar nuestro proyecto creando un repositorio llamado “primer\_repositorio”.

