

Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

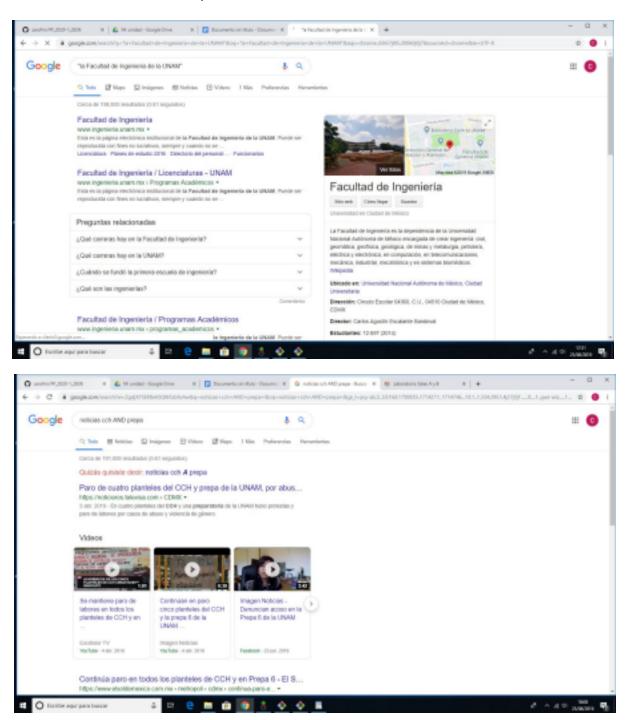
Laboratorios de computación salas A y B

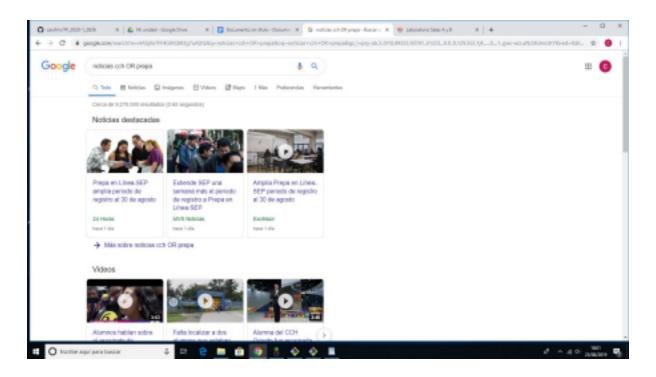
Profesor:	Alejandro Esteban Pimentel Alarcon	
Asignatura:	Fundamentos de Programación	
Grupo:	Bloque 135 3	
No de Práctica(s):	1°	
Integrante(s):	Franco Inglés Carolina	
No. de Equipo de cómputo empleado:		
No. de Lista o Brigada:	2836	
Semestre:	1°	
Fecha de entrega:	26 de agosto de 2019	
Observaciones:	Te recomiendo usar los comandos y las descripciones junto a las imágenes para recordar a qué te refieres.	
	En cuanto al reporte de git, te recomendaría separar claramente los comandos, serán mucho más legibles y claros	

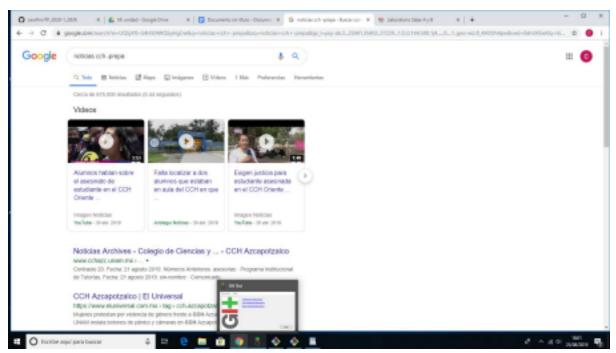
CALIFICACIÓN:	10

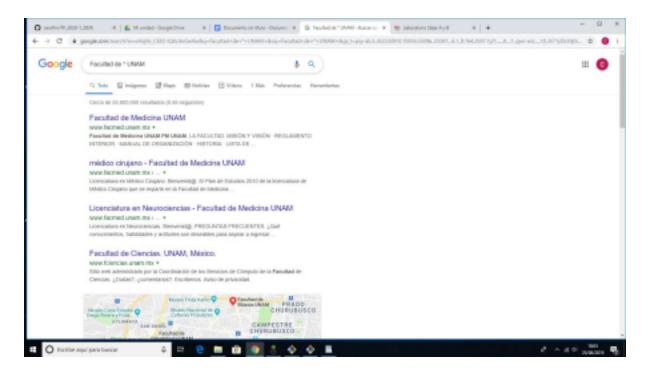
PRÁCTICA #1

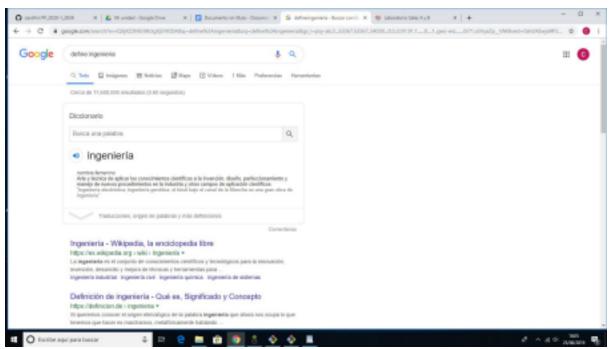
CAPTURAS DE PANTALLA (EJEMPLOS Y 3 COMBINACIONES:

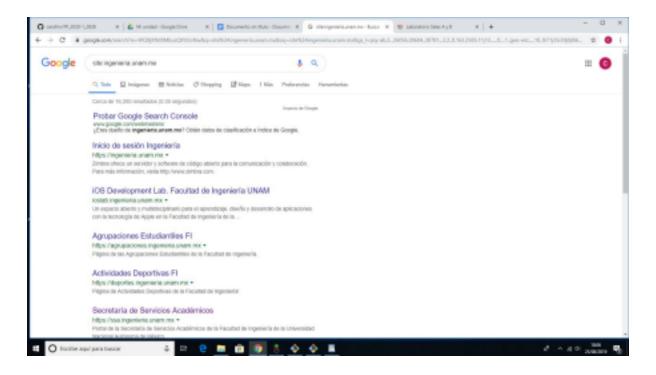


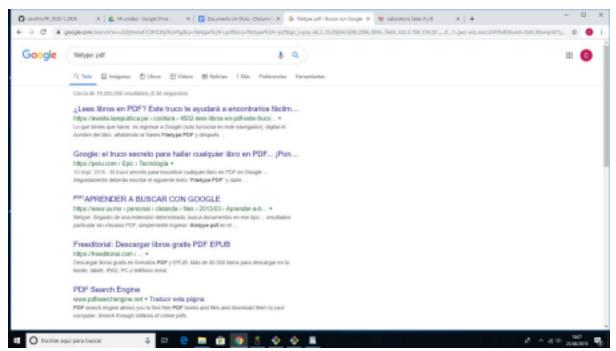


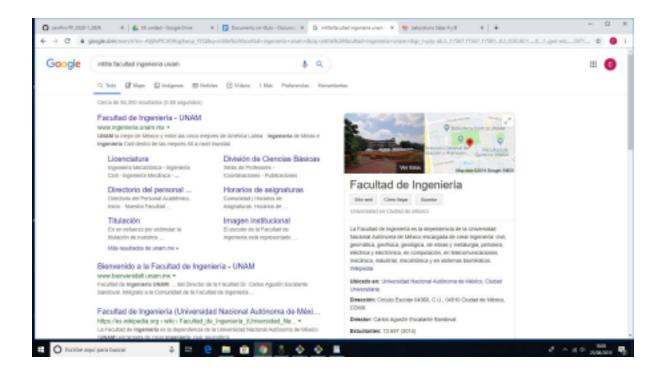


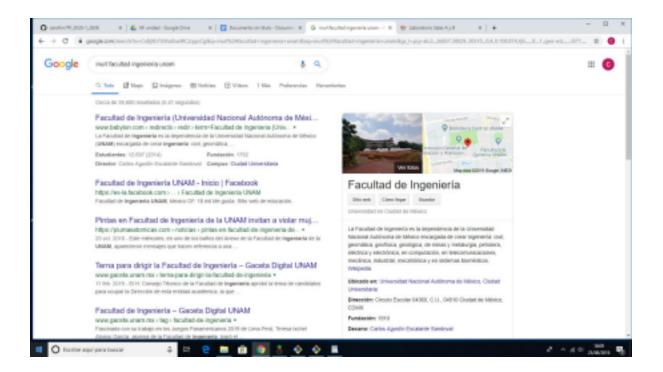


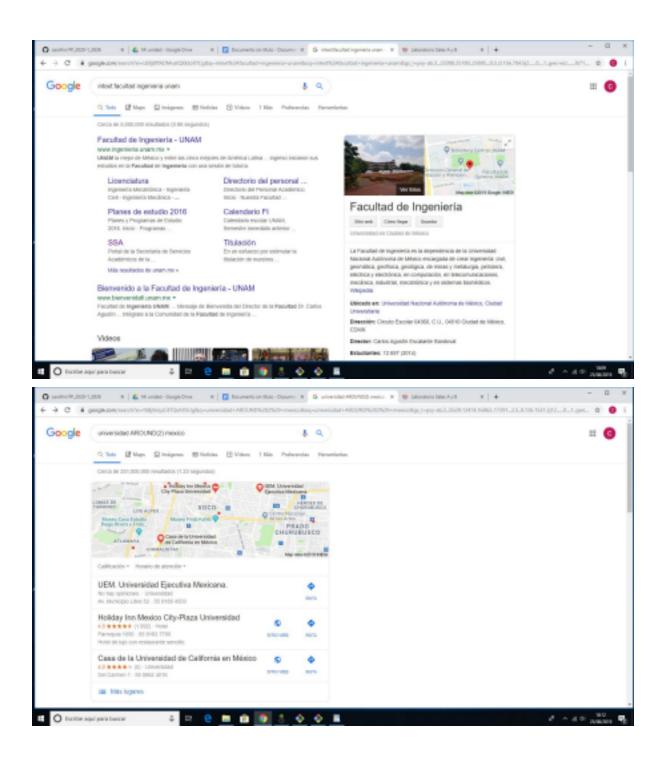


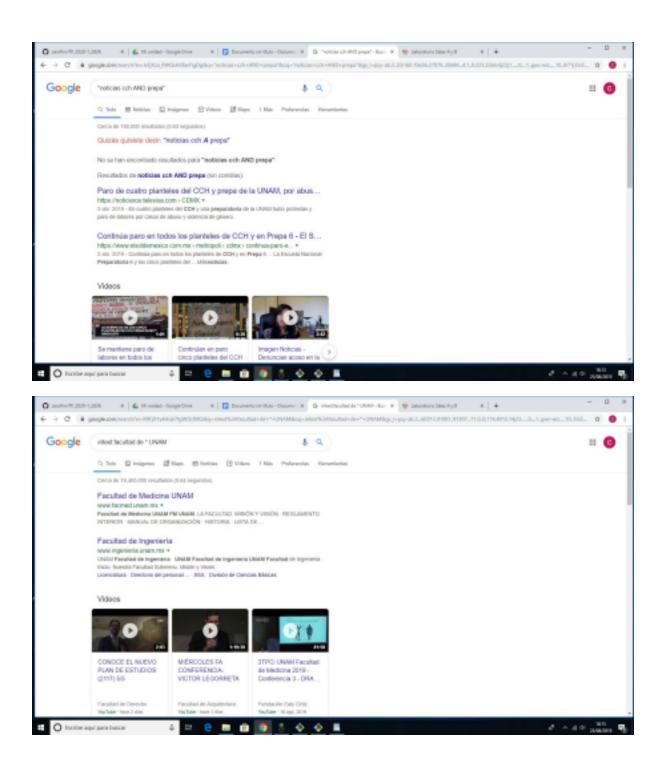


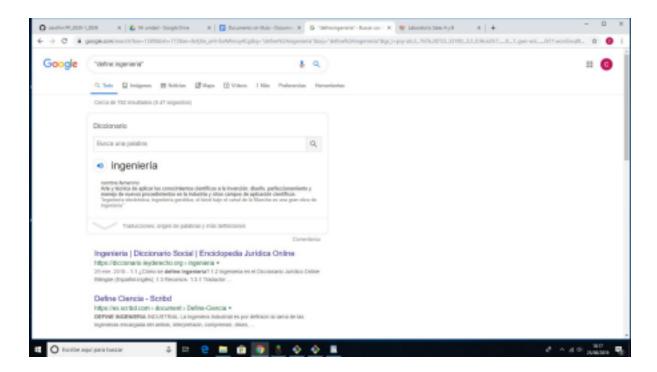




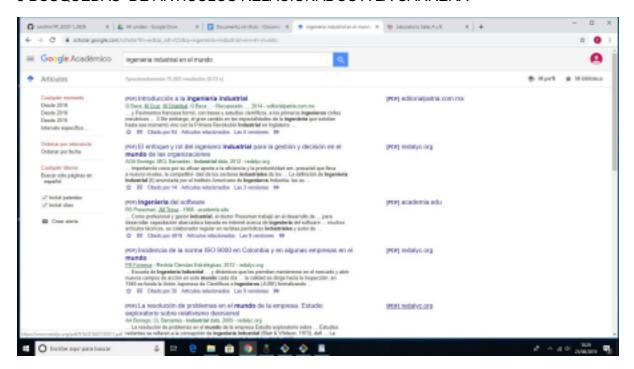


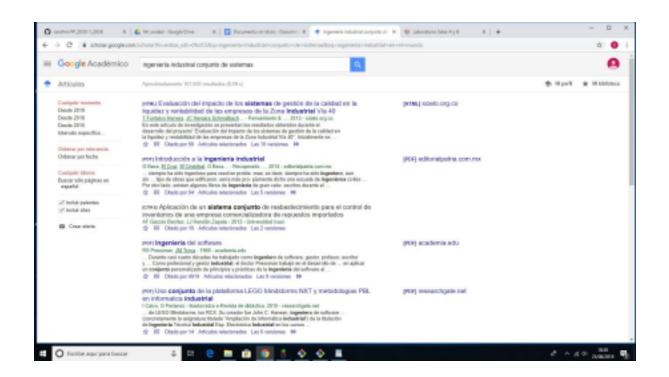


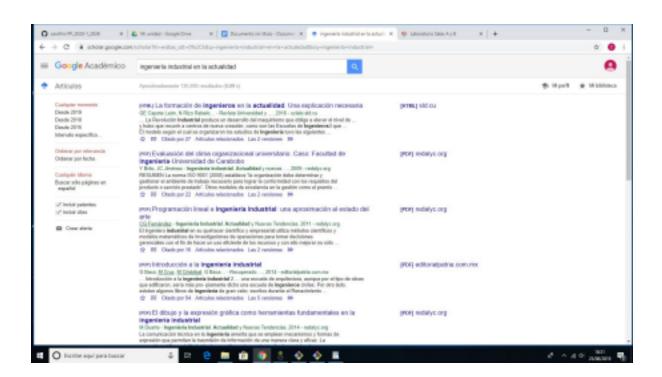


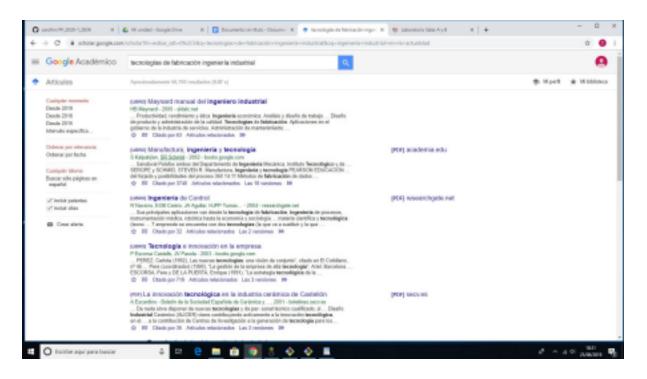


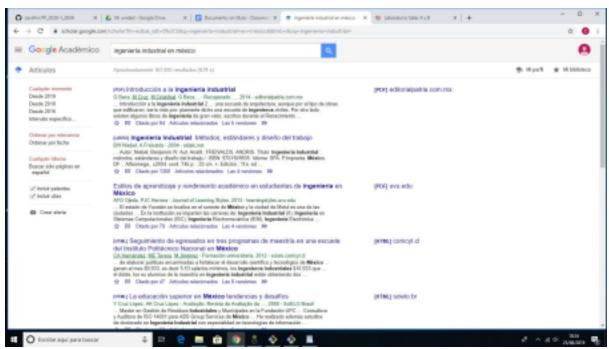
5 BÚSQUEDAS DE ARTÍCULOS RELACIONADOS A LA CARRERA



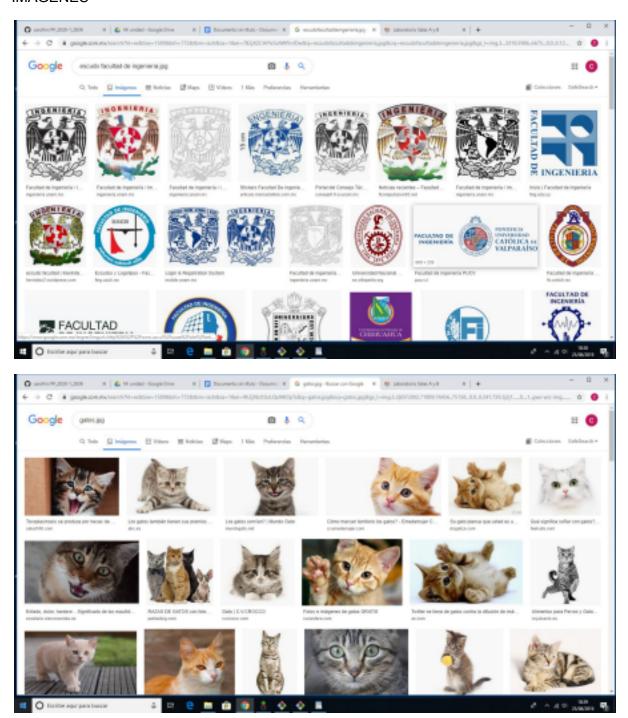








IMÁGENES



REPORTE PRÁCTICA #1

En la práctica #1 descubrimos a utilizar otras herramientas que google nos ofrece y nos enfocamos en el ámbito académico para mejorar las búsquedas que realizamos cotidianamente, poder perfeccionarlas y llevarlas a cabo correctamente.

Por otro lado en la segunda parte de la práctica comenzamos a trabajar en GIT (controlador de versiones) para poder hacer una cuenta en git hub en la cual vamos a guardar todas nuestras prácticas y tareas durante el semestre.

- Comenzamos con la configuración inicial identificando al programador (en este caso yo) con ayuda de nuestro user name e email.
- Iniciamos nuestro proyecto creando un directorio con mkdir llamado PROYECTO y git init esto crea una carpeta oculta en la que se va almacenar el seguimiento del proyecto.
- Revisamos el estado de los archivos git status para poder escribir con ayuda de echo "Mi primer proyecto" en el archivo README y lo añadimos con git add README.
- Modificamos el archivo README para ver el seguimiento con git status, git add README y de nuevo git status.
- Probamos el meter muchos archivos porque así se puede añadir la carpeta completa touch archivo1, touch archivo2, touch archivo3, etc. También es posible quererle dar seguimiento a todo excepto a algunos archivos en especial y para eso usamos .gitignore
- Supimos que el commit es la forma en la que se guarda el estado actualmente como una versión pero todos los commits necesitan un mensaje que generalmente es para anotar los cambios que se han realizado. El mensaje commit se guarda con los cambios que realizamos en el proyecto.
- Posteriormente borramos y movimos algunos archivos para aprender a cómo hacerlo en un futuro es solo a manera de práctica; con la opción cached el archivo temp se saca del seguimiento pero no se elimina de la carpeta.
- Es común cometer errores y olvidar guardar los cambios del proyecto y para esto utilizamos mv temp no_temp, git add ., git commit -m "Siempre si quiero agregar el archivo temporal", echo "Olvide editar mi nuevo archivo!" >> no_temp, git add ., git commit --amend gracias a esto se abre un editor en el cual puedes cambiar el mensaje ya indicado antes.
- También ejecutando diferentes instrucciones como git reset HEAD podemos ir a versiones anteriores por si cometimos algún error o si se nos olvidó algo y con git log podemos descartar cambios hasta puntos específicos e igual podemos regresar archivos a como estaban en un principio borrando y moviendo archivos.
- Para finalizar creamos una cuenta en git hub junto con un repositorio dentro de esta, copiamos la url del repositorio en la terminal para clonarlo.