

NOMBRE DE LA PRÁCTICA	INTRODUCCIÓN AL LENGUAJE C.			NO.	1
ASIGNATURA:	MÉTODOS NUMÉRICOS.	CARRERA:	INGENIERÍA EN SISTEMAS COMPUTACIONALES	DURACIÓN DE LA PRÁCTICA (HRS)	2 HORAS

**NOMBRE DEL ALUMNO:** Jorge Miranda Zuniga

**GRUPO:** 3402

## I. Competencia(s) específica(s):

- Aplica los tipos de errores para identificar la incertidumbre y limitaciones de los cálculos numéricos en una computadora.

Encuadre con CACEI: Registra el (los) atributo(s) de egreso y los criterios de desempeño que se evaluarán en esta práctica.

No. atributo	Atributos de egreso del PE que impactan en la asignatura
1	Reconocer la necesidad permanente de conocimiento adicional y tener la habilidad para localizar, evaluar, integrar y aplicar este conocimiento adecuadamente.

## II. Lugar de realización de la práctica (laboratorio, taller, aula u otro):

- Casa

## III. Material empleado:

- Computadora
- Visual Studio Code
- Git
- Terminal

## IV. Desarrollo de la práctica:

- 1) Lo primero que vamos a hacer será instalar git, para esto abrimos una terminal y copiamos la siguiente línea **sudo apt-get install git**, ingresamos nuestra contraseña, después de esto esperamos a que se instale.

```
jorge@TheCracker: ~  
jorge@TheCracker:~$ sudo apt-get install git  
[sudo] contraseña para jorge:  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
git ya está en su versión más reciente (1:2.25.1-1ubuntu3.11).  
Los paquetes indicados a continuación se instalaron de forma automática y ya no  
son necesarios.  
adwaita-qt chromium-codecs-ffmpeg-extra coinor-libcbc3 coinor-libcgl1  
coinor-libclp1 coinor-libcoinmp1v5 coinor-libcoinutils3v5 coinor-libosi1v5  
gstreamer1.0-vaapi heif-gdk-pixbuf libdouble-conversion3 libpcre2-16-0  
libqt5core5a libqt5dbus5 libqt5gui5 libqt5network5 libqt5svg5 libqt5widgets5  
libva-wayland2 libxcb-xinerama0 libxcb-xinput0 libxmlsec1 libxmlsec1-nss  
qt5-gtk-platformtheme qttranslations5-l10n  
Utilice «sudo apt autoremove» para eliminarlos.  
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 3 no actualizados.  
jorge@TheCracker:~$
```

- 2) Ahora nos iremos a nuestro navegador y crearemos un nuevo repositorio en nuestra cuenta de GitHub, como nombre le pondremos algo relacionado a la materia, marcamos la opción de **public** y lo demás lo dejamos como esta por defecto.

### Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner \* Jorge186414 / Repository name \* CodigosMetodosNumericos  
CodigosMetodosNumericos is available.

Great repository names are short and memorable. Need inspiration? How about [vigilant-octo-engine?](#)

Description (optional)

☒ **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**  
You choose who can see and commit to this repository.

Initialize this repository with:  
☐ **Add a README file**  
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore  
.gitignore template: None  
Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license  
License: None  
A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

☒ You are creating a public repository in your personal account.

[Create repository](#)



- 3) Como cambio la configuración de GitHub tenemos que generar un token, para esto nos vamos a los ajustes de nuestro perfil y buscamos los ajustes para desarrolladores, damos clic en **personal access tokens** y luego generamos el mismo. Para esto nos pedirá ingresar nuestra contraseña la ingresamos. Nos pedirá una nota o nombre en mi caso colocare el nombre de la materia, y para finalizar marcamos todas las casillas para evitar problemas al momento de intentar subir los documentos al repositorio en GitHub.

## Edit personal access token (classic)

If you've lost or forgotten this token, you can regenerate it, but be aware that any scripts or applications using this token will need to be updated.

[Regenerate token](#)

### Note

What's this token for?

### Expiration

This token expires *on Fri, Jun 7 2024*. To set a new expiration date, you must [regenerate the token](#).

### Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

<input checked="" type="checkbox"/> <b>repo</b>	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events
<input checked="" type="checkbox"/> <b>workflow</b>	Update GitHub Action workflows
<input checked="" type="checkbox"/> <b>write:packages</b>	Upload packages to GitHub Package Registry
<input checked="" type="checkbox"/> read:packages	Download packages from GitHub Package Registry
<input checked="" type="checkbox"/> <b>delete:packages</b>	Delete packages from GitHub Package Registry



<input checked="" type="checkbox"/> <b>admin:org</b> <input checked="" type="checkbox"/> write:org <input checked="" type="checkbox"/> read:org <input checked="" type="checkbox"/> manage_runners:org	Full control of orgs and teams, read and write org projects Read and write org and team membership, read and write org projects Read org and team membership, read org projects Manage org runners and runner groups
<input checked="" type="checkbox"/> <b>admin:public_key</b> <input checked="" type="checkbox"/> write:public_key <input checked="" type="checkbox"/> read:public_key	Full control of user public keys Write user public keys Read user public keys
<input checked="" type="checkbox"/> <b>admin:repo_hook</b> <input checked="" type="checkbox"/> write:repo_hook <input checked="" type="checkbox"/> read:repo_hook	Full control of repository hooks Write repository hooks Read repository hooks
<input checked="" type="checkbox"/> <b>admin:org_hook</b>	Full control of organization hooks
<input checked="" type="checkbox"/> <b>gist</b>	Create gists
<input checked="" type="checkbox"/> <b>notifications</b>	Access notifications
<input checked="" type="checkbox"/> <b>user</b> <input checked="" type="checkbox"/> read:user <input checked="" type="checkbox"/> user:email <input checked="" type="checkbox"/> user:follow	Update ALL user data Read ALL user profile data Access user email addresses (read-only) Follow and unfollow users
<input checked="" type="checkbox"/> <b>delete_repo</b>	Delete repositories
<input checked="" type="checkbox"/> <b>write:discussion</b> <input checked="" type="checkbox"/> read:discussion	Read and write team discussions Read team discussions
<input checked="" type="checkbox"/> <b>admin:enterprise</b> <input checked="" type="checkbox"/> manage_runners:enterprise <input checked="" type="checkbox"/> manage_billing:enterprise <input type="checkbox"/> read:enterprise	Full control of enterprises Manage enterprise runners and runner groups Read and write enterprise billing data Read enterprise profile data
<input checked="" type="checkbox"/> <b>audit_log</b> <input checked="" type="checkbox"/> read:audit_log	Full control of audit log Read access of audit log
<input checked="" type="checkbox"/> <b>codespace</b> <input checked="" type="checkbox"/> codespace:secrets	Full control of codespaces Ability to create, read, update, and delete codespace secrets

<input checked="" type="checkbox"/> project	Full control of projects
<input checked="" type="checkbox"/> read:project	Read access of projects
<input checked="" type="checkbox"/> admin:pgp_key	Full control of public user GPG keys
<input checked="" type="checkbox"/> write:pgp_key	Write public user GPG keys
<input checked="" type="checkbox"/> read:pgp_key	Read public user GPG keys
<input checked="" type="checkbox"/> admin:ssh_signing_key	Full control of public user SSH signing keys
<input checked="" type="checkbox"/> write:ssh_signing_key	Write public user SSH signing keys
<input checked="" type="checkbox"/> read:ssh_signing_key	Read public user SSH signing keys

- 4) Ya que instalamos git, creamos nuestro repositorio y configuramos nuestro token, vamos a configurar nuestras credenciales en la terminal usaremos la siguiente línea de código **git config --global user.TU\_NOMBRE\_DE\_USUARIO** y **git config --global user.TU\_CORREO** esto para poder hacer las peticiones y solicitudes con git.

```
jorge@TheCracker: ~  
jorge@TheCracker:~$ git config --global user.name Jorge186414  
jorge@TheCracker:~$ git config --global user.email jorgaismiranda@gmail.com  
jorge@TheCracker:~$ git config user.email jorgaismiranda@gmail.com  
jorge@TheCracker:~$ git config user.name  
Jorge186414  
jorge@TheCracker:~$ git config user.email  
jorgaismiranda@gmail.com  
jorge@TheCracker:~$
```

- 5) Ahora en la terminal navegamos hasta la ruta donde tenemos los archivos que queremos subir, y ya que estamos en la ruta escribimos **git init**.

```
jorge@TheCracker: ~/PracticasZamora/RepositorioGitHub  
jorge@TheCracker:~/PracticasZamora/RepositorioGitHub$ git init  
Iniciado repositorio Git vacío en /home/jorge/PracticasZamora/RepositorioGitHub/.git/  
jorge@TheCracker:~/PracticasZamora/RepositorioGitHub$
```



- 6) Ahora escribimos **git add** . Esto para que se suban todos los archivos dentro de la carpeta. Seguido de esto hacemos un comit para poner como una descripción cuando se suba para esto escribimos **git commit -m "Codigos En C Metodos Numericos"**

```
jorge@TheCracker:~/Practicazamora/RepositorioGitHub$ git add .
jorge@TheCracker:~/Practicazamora/RepositorioGitHub$ git commit -m "Codigos en C Metodos Numericos"
[master (commit-raiz) 04ffca7] Codigos en C Metodos Numericos
102 files changed, 2768 insertions(+)
create mode 100644 10_Arreglos/46_arreglo1.c
create mode 100755 10_Arreglos/47_arreglo2.c
create mode 100755 10_Arreglos/48_arreglo3.c
create mode 100755 10_Arreglos/49_arreglo4.c
create mode 100755 10_Arreglos/50_arreglo5.c
create mode 100755 10_Arreglos/51_arreglo6.c
create mode 100755 10_Arreglos/52_arreglo7.c
create mode 100644 11_For/53_for1.c
create mode 100755 11_For/54_for2.c
create mode 100755 11_For/55_for3.c
create mode 100755 11_For/56_for4.c
create mode 100644 11_For/57_for5.c
create mode 100755 11_For/58_for6.c
create mode 100755 11_For/59_for7.c
create mode 100755 12_Cadenas/60_cadena1.c
create mode 100644 12_Cadenas/61_cadena2.c
create mode 100755 12_Cadenas/62_cadena3.c
create mode 100755 12_Cadenas/63_cadena4.c
create mode 100755 12_Cadenas/64_cadena5.c
create mode 100755 12_Cadenas/65_cadena6.c
create mode 100755 12_Cadenas/66_cadena7.c
create mode 100755 12_Cadenas/67_cadena8.c
create mode 100755 12_Cadenas/69_cadena9.c
create mode 100755 12_Cadenas/70_cadena10.c
create mode 100755 12_Cadenas/71_cadena11.c
create mode 100755 12_Cadenas/72_cadena12.c
create mode 100755 13_Funciones/73_funcion1.c
create mode 100755 13_Funciones/74_funcion2.c
create mode 100755 13_Funciones/75_funcion3.c
create mode 100755 13_Funciones/76_funcion4.c
create mode 100644 13_Funciones/77_funcionRecursiva1.c
create mode 100644 13_Funciones/78_funcionesRecursivas2.c
create mode 100755 14_Estructuras/79_estructuras1.c
create mode 100755 14_Estructuras/80_estructuras2.c
create mode 100755 14_Estructuras/81_estructuras3.c
create mode 100755 14_Estructuras/82_estructuras4.c
create mode 100755 14_Estructuras/83_estructuras5.c
create mode 100755 14_Estructuras/84_estructuras6.c
create mode 100755 15_Apuntadores/85_apuntador1.c
create mode 100755 15_Apuntadores/86_apuntadorEjem.c
create mode 100755 15_Apuntadores/87_apuntadorEjem2.c
create mode 100755 15_Apuntadores/88_codigoEjApuntador.c
create mode 100755 15_Apuntadores/89_size0f1.c
create mode 100644 15_Apuntadores/90_size0f2.c
create mode 100755 15_Apuntadores/91_reservacionDinamica.c
create mode 100755 15_Apuntadores/92_reservacionDinamica2.c
create mode 100755 15_Apuntadores/93_direccionesDeArreglo.c
create mode 100755 15_Apuntadores/94_direccionesDeArreglo2.c
```





```
create mode 100755 15_Apuntadores/95_direccionesDeArreglo3.c
create mode 100755 16_Tareas/100_reproWav.c
create mode 100755 16_Tareas/101_wavwrite.c
create mode 100755 16_Tareas/102_ciclosReloj.c
create mode 100755 16_Tareas/103_tareaArquitectura.c
create mode 100755 16_Tareas/96_convertirMP3.c
create mode 100755 16_Tareas/97_crearWav.c
create mode 100755 16_Tareas/98_generarOndas.c
create mode 100755 16_Tareas/99_reproducirwav.c
create mode 100755 2_LenguajeC/1_helloWorld.c
create mode 100755 2_LenguajeC/2_printFormatted.c
create mode 100755 2_LenguajeC/3_ejemplo1.c
create mode 100755 2_LenguajeC/4_ejemplo2.c
create mode 100755 2_LenguajeC/5_operacionesAritmeticas.c
create mode 100755 3_VariablesC/10_variables4.c
create mode 100755 3_VariablesC/11_variables5.c
create mode 100755 3_VariablesC/12_variables6.c
create mode 100755 3_VariablesC/13_variables7.c
create mode 100755 3_VariablesC/14_variables8.c
create mode 100755 3_VariablesC/6_valorPI.c
create mode 100755 3_VariablesC/7_variables1.c
create mode 100755 3_VariablesC/8_variables2.c
create mode 100755 3_VariablesC/9_variables3.c
create mode 100755 4_Operador_Incremento/15_postDecreIncre.c
create mode 100755 4_Operador_Incremento/16_postPredDecreIncre.c
create mode 100755 4_Operador_Incremento/17_preinDeIn.c
create mode 100755 4_Operador_Incremento/18_multiplicacionPostPre.c
create mode 100755 5_PrecedenciaOperadores/19_precedencia1.c
create mode 100755 5_PrecedenciaOperadores/20_precedencia2.c
create mode 100755 5_PrecedenciaOperadores/21_precedencia3.c
create mode 100755 6_OperadoresLogicosRelacion/22_operadorLogico1.c
create mode 100755 6_OperadoresLogicosRelacion/23_operadorLogico2.c
create mode 100755 6_OperadoresLogicosRelacion/24_operadorLogico3.c
create mode 100755 6_OperadoresLogicosRelacion/25_operadorLogico4.c
create mode 100644 8_If/26_if1.c
create mode 100755 8_If/27_if1.c
create mode 100644 8_If/28_if2.c
create mode 100755 8_If/29_if3.c
create mode 100644 8_If/30_ifAnidado1.c
create mode 100755 8_If/31_ifAnidado2.c
create mode 100644 8_If/32_ifAnidados3.c
create mode 100755 8_If/33_ifAnidado4.c
create mode 100755 8_If/34_ifElse1.c
create mode 100755 8_If/35_ifElse2.c
create mode 100755 8_If/36_ifElse3.c
create mode 100755 8_If/37_operadorCondicional1.c
create mode 100755 9_While/38_while1.c
create mode 100755 9_While/39_while2.c
create mode 100755 9_While/40_while3.c
create mode 100755 9_While/41_while4.c
create mode 100644 9_While/42_while5.c
create mode 100644 9_While/43_while6.c
create mode 100644 9_While/44_while7.c
create mode 100755 9_While/45_doWhile1.c
```

ianpa@ThePacker:~/Practicas7amano/DepositorioGitHub



- 7) Añadimos el repositorio de forma remota copiando el token que generamos con anterioridad lo colocamos antes de nuestro nombre de usuario el, la liga al repositorio la podemos copiar desde este mismo.

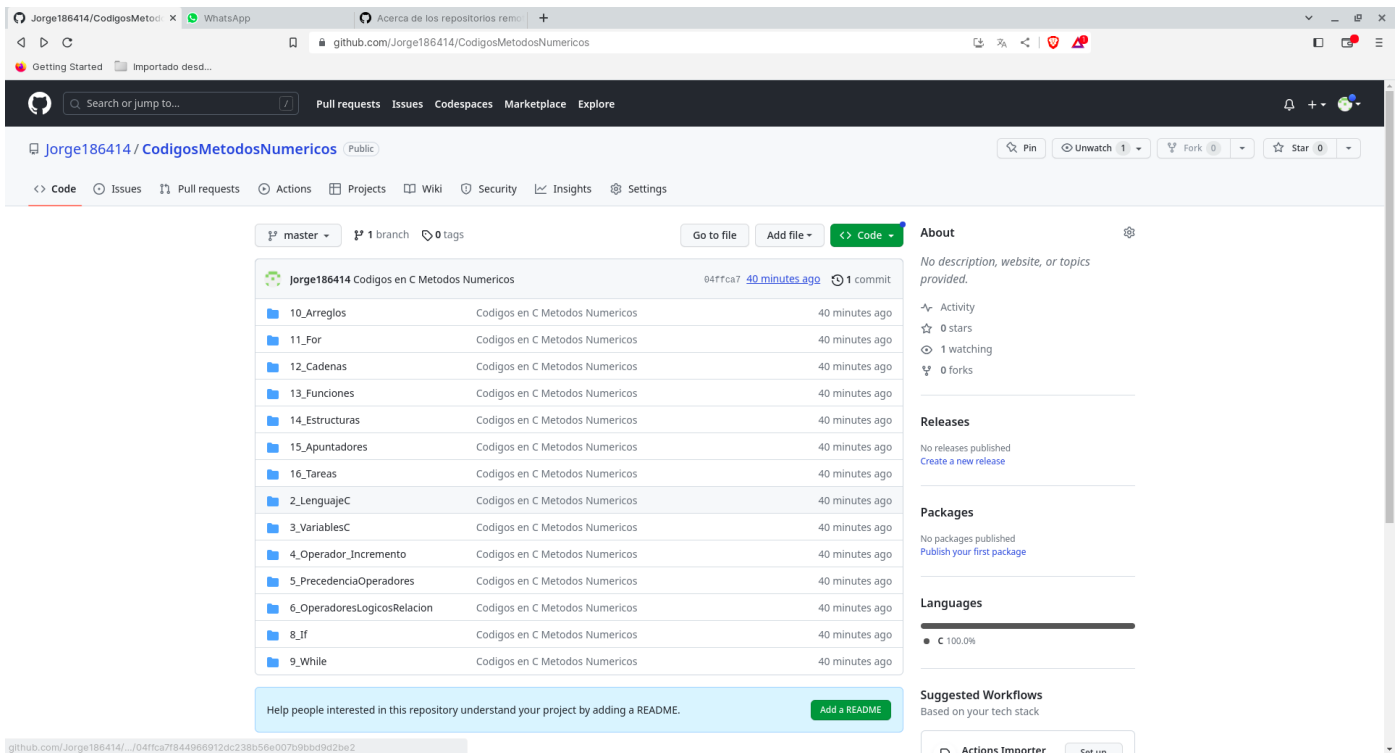
```
jorge@TheCracker:~/PracticasZamora/RepositorioGitHub$ git remote add origin https://ghp_EkZuqKwA06qoWXZnrQYWMrkkBYLQcM03DPaY@github.com/Jorge186414/CodigosMetodosNumericos
```

- 8) Ahora tenemos que subir los archivos al repositorio para esto usaremos la siguiente linea **git push -u origin master** la palabra master se refiere a la rama de nuestro repositorio, esta puede cambiar si creamos una nueva rama, pero por lo común es master o main. Una vez que hagamos esto se comenzaran a subir los archivos al repositorio.

```
jorge@TheCracker:~/PracticasZamora/RepositorioGitHub$ git push -u origin master
Enumerando objetos: 117, listo.
Contando objetos: 100% (117/117), listo.
Compresión delta usando hasta 4 hilos
Comprimiendo objetos: 100% (117/117), listo.
Escribiendo objetos: 100% (117/117), 25.78 KiB | 1.72 MiB/s, listo.
Total 117 (delta 9), reusado 0 (delta 0)
remote: Resolving deltas: 100% (9/9), done.
To https://github.com/Jorge186414/CodigosMetodosNumericos
 * [new branch]      master -> master
Rama 'master' configurada para hacer seguimiento a la rama remota 'master' de 'origin'.
jorge@TheCracker:~/PracticasZamora/RepositorioGitHub$
```



- 9) Para comprobar la subida de archivos nos vamos a nuestro navegador y abrimos nuestro repositorio donde encontraremos todos los archivos de nuestra carpeta antes de subirlos.

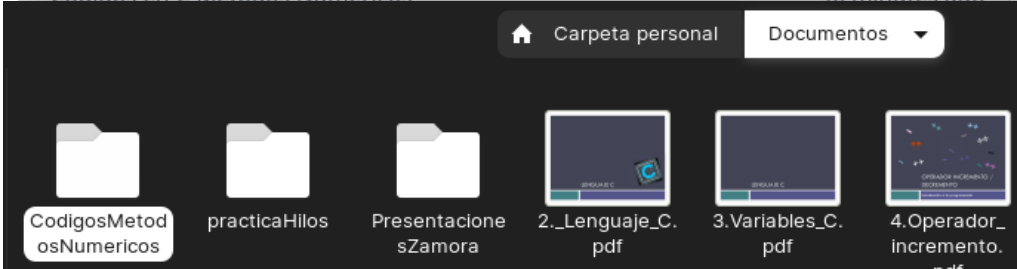


- 10) Para clonar el repositorio en nuestro equipo primero navegaremos hasta la carpeta donde queramos que se clone, ya que estamos ahí **escribimos git clone URL\_del\_repositorio**. En automático esto clonará el repositorio para poder visualizarlo en nuestra computadora.

```
jorge@TheCracker: ~/Documentos
jorge@TheCracker:~$ cd Documentos/
jorge@TheCracker:~/Documentos$ ls
'10. Arreglos.pdf'
'11. For.pdf'
'12. Cadenas.pdf'
'13. Funciones.pdf'
'14. Estructuras.pdf'
'15. Apuntadores.pdf'
2._Lenguaje_C.pdf
3.Variables_C.pdf
4.Operador_incremento.pdf
5.Precendencia_de_operadores.pdf
6.Operadores_logicos_y_de_relación.pdf
'7. Entrada y Salida de datos.pdf'
'8. If.pdf'
'9. While.pdf'
Buses.odp
iptables-desde-cero.pdf
'Manual de Practicas.pdf'
ManualdePracticasGitHub.odt
ManualdePracticas.odt
ManualDePracticasSubidaGitHub.odt
practicaMitos
practicaNormalizacion.odt
PresentacionesZamora
```



```
jorge@TheCracker:~/Documentos$ git clone https://github.com/Jorge186414/CodigosMetodosNumericos.git
Clonando en 'CodigosMetodosNumericos'...
remote: Enumerating objects: 117, done.
remote: Counting objects: 100% (117/117), done.
remote: Compressing objects: 100% (108/108), done.
remote: Total 117 (delta 9), reused 117 (delta 9), pack-reused 0
Recibiendo objetos: 100% (117/117), 25.78 KiB | 8.59 MiB/s, listo.
Resolviendo deltas: 100% (9/9), listo.
jorge@TheCracker:~/Documentos$ ls
'10. Arreglos.pdf'      '8. If.pdf'
'11. For.pdf'          '9. While.pdf'
'12. Cadenas.pdf'      Buses.odp
'13. Funciones.pdf'    CodigosMetodosNumericos
'14. Estructuras.pdf' iptables-desde-cero.pdf
'15. Apuntadores.pdf' 'Manual de Practicas.pdf'
2._Lenguaje_C.pdf      ManualdePracticasGitHub.odt
3.Variables_C.pdf      ManualdePracticas.odt
4.Operador_incremento.pdf ManualDePracticasSubidaGitHub.odt
5.Precedencia_de_operadores.pdf practicaHilos
6.Operadores_logicos_y_de_relación.pdf practicaNormalizacion.ods
'7. Entrada y Salida de datos.pdf' PresentacionesZamora
jorge@TheCracker:~/Documentos$
```



**Conclusiones:** El saber como subir archivos a un repositorio en GitHub nos es de ayuda para mantener respaldados nuestros archivos ya nos hace un recopilatorio de cambios que hayamos hecho en algún archivo que subimos al repositorio permitiéndonos recuperar código en caso de que nos hayamos equivocado y queramos regresar a una versión anterior con esto también tenemos evidencia de que si hicimos un trabajo en tiempo y forma.