

# Tarea #2

Inyerman Alexander Xap Chin  
20210097

Escuela de Mecánica Eléctrica,  
Facultad de Ingeniería, Universidad  
de San Carlos de Guatemala.

## A. Resumen

En esta actividad se instaló tanto Visual Studio Code como GitHub. Primeramente, se descargó Visual Studio Code y se instaló.

Luego se descargó GitHub y mientras lo instalábamos nos aseguramos de elegir como editor predeterminado Visual Studio Code que ya habíamos descargado anteriormente. Además de loguearnos y sincronizarlo con nuestra carpeta donde guardamos nuestros códigos, escribiendo los códigos que nos da el mismo GitHub pudimos preparar y sincronizar nuestra carpeta. Luego de una verificación de nuestro correo pudimos ver que todo estaba realizado correctamente por lo que procedimos a añadir a nuestro ingeniero a cargo como colaborador.

Y por último creamos nuestros códigos verificamos que todo se ejecutara correctamente y los subimos al repositorio.

## B. Objetivos

- Crear un repositorio remoto utilizando GitHub para nuestra carpeta de códigos y

aprender sobre la librería `stdlib.h`.

- Aprender los diferentes comandos que ofrece la librería `stdlib.h`
- Guardar nuestros códigos pasados y los que realizaremos en el futuro en nuestro repositorio.
- Crear dos códigos sencillos que muestren lo básico de la librería `stdlib.h`
- Agregar como colaborador al ingeniero a cargo del curso para que pueda verificar nuestros códigos y brindar retroalimentación.
- Aprender a subir nuestros códigos al repositorio, así como bajar las actualizaciones del código.

## C. Marco Teórico

### GitHub:

Es una plataforma donde podemos almacenar, compartir y trabajar junto con otros usuarios para escribir código.

Utilizar esta plataforma nos permite:

- **Presentar o compartir** el trabajo.
- **Seguir y administrar** los cambios en el código a lo largo del tiempo.
- Dejar que otros usuarios **revisen** el código y realicen sugerencias para mejorarlo.
- **Colaborar** en un proyecto compartido, sin preocuparse

de que los cambios afectarán al trabajo de los colaboradores antes de que esté listo para integrarlos.

El trabajo colaborativo, una de las características fundamentales de GitHub, es posible gracias al software de código abierto Git, en el que se basa GitHub.

### **Git:**

Git es un sistema de control de versiones que realiza un seguimiento de los cambios en los archivos. Git es especialmente útil cuando un grupo de personas y tú estáis haciendo cambios en los mismos archivos al mismo tiempo. ¿Cómo funcionan Git y GitHub juntos?

Al cargar archivos en GitHub, los almacenas en un "repositorio de Git". Esto significa que al realizar cambios (o "compromisos") en los archivos de GitHub, Git se iniciará automáticamente para realizar un seguimiento y administrar los cambios.

Hay muchas acciones relacionadas con Git que puedes completar en GitHub directamente en el navegador, como crear un repositorio de Git, crear ramas y cargar y editar archivos.

Sin embargo, la mayoría de las personas trabajan en sus archivos localmente (en su propio ordenador), luego sincronizan continuamente estos cambios locales y todos los datos de Git relacionados, con el repositorio central "remoto" en

GitHub. Hay muchas herramientas que puedes usar para hacerlo, como GitHub Desktop.

Cuando empieces a colaborar con otros y todos necesitéis trabajar en el mismo repositorio al mismo tiempo, haréis lo siguiente continuamente:

Extraer todos los cambios más recientes realizados por los colaboradores del repositorio remoto en GitHub.

Transferir de vuelta los cambios al mismo repositorio remoto en GitHub.

Git determina cómo combinar inteligentemente este flujo de cambios y GitHub te ayuda a administrar el flujo a través de

características como "solicitudes de cambios".

### **La Librería `stdlib.h`:**

La librería `stdlib.h` contiene muchas utilidades para los programadores, tales como la gestión de memoria dinámica, la ordenación y búsqueda en arrays o el control de procesos.

La **especificación** de la librería es la siguiente:

#### **Conversiones numéricas.**

`atof`: Convierte una cadena de caracteres a `float`.

`atoi`: Convierte una cadena de caracteres a `int`.

`atol`: Convierte una cadena de caracteres a `long`.

`strtod`: Convierte una cadena de caracteres a `double`.

strtol: Convierte una cadena de caracteres a long.

strtoul: Convierte una cadena de caracteres a unsigned long.

### **Funciones matemáticas.**

abs: Valor absoluto.

labs: Valor absoluto.

div: División entera.

ldiv: División entera.

### **Widechar conversions**

wcstombs: Convierte una cadena de caracteres anchos a una cadena multibyte.

wctomb: Convierte un carácter a un carácter multibyte.

### **Conversiones Multibyte**

mblen: Devuelve el número de bytes de un carácter multibyte.

mbstowcs: Convierte de cadena multibyte a cadena ancha.

mbtowc: Convierte de cadena multibyte a carácter.

### **Manipulación de memoria**

calloc: Reserva memoria dinámica.

free: Libera memoria dinámica.

realloc: Reserva memoria dinámica.

malloc: Reserva memoria dinámica.

### **Control de procesos**

abort: Termina la ejecución de forma anormal.

atexit: Registra una función callback para salir.

exit: Termina la ejecución del programa.

getenv: Recupera una variable de entorno.

system: Ejecuta un comando externo.

### **Generación de números aleatorios**

rand: Genera un número aleatorio.

srand: Inicializa el generador de números aleatorios.

### **Operadores**

sizeof: Calcula el tamaño de cualquier tipo de dato.

### **Utilidades para arrays**

bsearch: Realiza una búsqueda binaria.

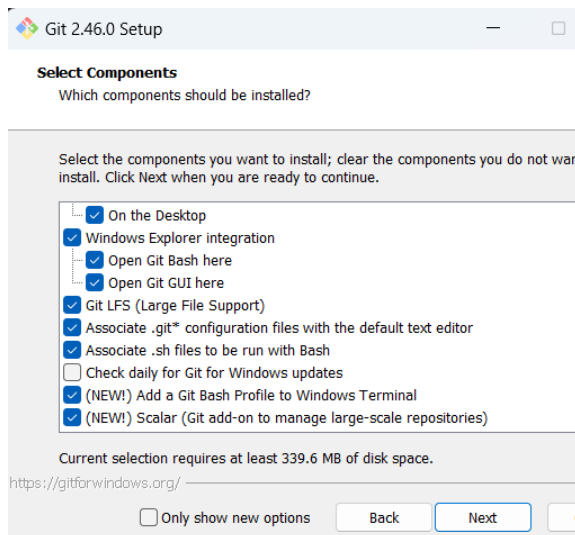
qsort: Ordena el array por Quicksort.

## **D. Marco Practico**

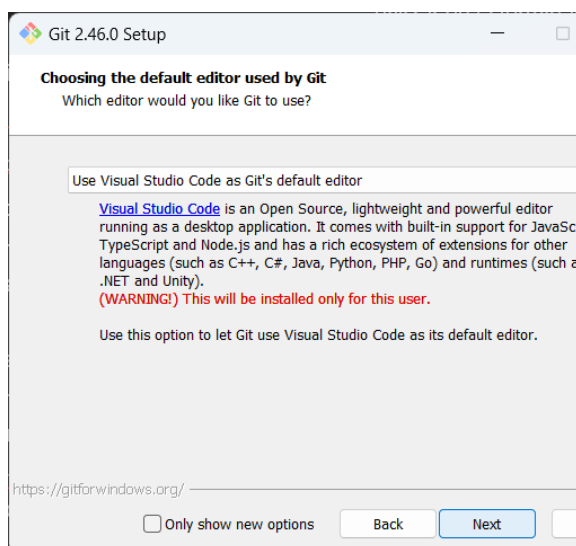
Lo primero que se realizó fue descargar VisualStudioCode, luego lo instalamos.

Luego descargamos GitHub y lo instalamos.

Luego le damos que si se pueda utilizar en la terminal en los pasos de instalación y agregar en el escritorio.



Luego le decimos que queremos que nuestro editor por defecto sea Visual Studio Code que instalamos previamente.



Luego procedemos a dejar todo como esta y dar siguiente hasta instalarlo.

Luego vamos a GitHub en nuestro navegador y procedemos a crearnos una cuenta.

Luego abrimos el programa GitHub y procedemos a registrar nuestro nombre de usuario y email, con los comandos

```
$ git config --global user.name "nombre de usuario"
```

Y

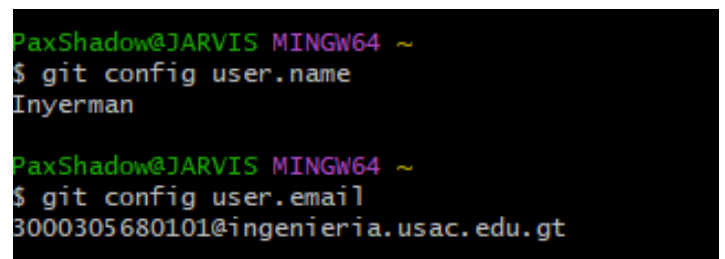
```
$ git config --global user.email "correo"
```

Y para corroborar se utiliza el código

```
$ git config user.name
```

Y

```
$ git config user.email
```



Luego vamos a nuestra carpeta que deseamos respaldar en el repositorio dentro de la terminal.

Y escribimos el código \$ git init

Luego de esto vamos a la pagina de GitHub y creamos un repositorio donde nos dará los códigos para poder vincular nuestra carpeta local al repositorio.

Procedemos a realizar los pasos siguientes en orden;

```
git init
```

```
git add .
```

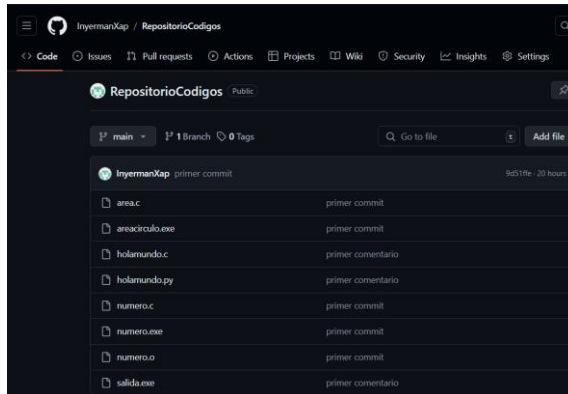
```
git commit -m "comentario personalizado"
```

```
git branch -M main
```

```
git remote add origin https://github.com/usuario/repositorio.git
```

```
git push -u origin main
```

Luego de hacer todo esto podemos ir a la pagina de github y comprobar que nuestros códigos que estaban en la carpeta se han subido correctamente.



Luego añadimos a nuestro colaborador que en este caso es el ingeniero a cargo del curso.

Le damos click en settings y en colaboradores y a añadir personas y agregamos el email dado por el ingeniero.

Luego realizamos los códigos correspondientes a la tarea #2 y luego de comprobar su funcionamiento.

Lo subimos a nuestro repositorio.

## E. Código

Parte 1/2 de la tarea#2 en lenguaje c

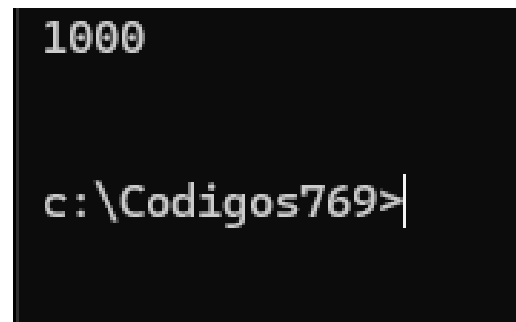


Parte 2/2 de la tarea#2 en lenguaje c

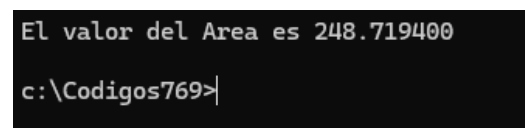


## F. Resultados

Compilamos y corremos la primera parte de nuestra tarea y corre sin problemas.



Luego el segundo también lo compilamos y corremos y vemos que corre sin problema.



## G. Conclusiones

- Con la culminación de esta actividad hemos logrado crear con éxito un repositorio remoto en GitHub.
- Aprendimos varios comandos de la librería `stdlib.h`.
- Guardamos con éxito nuestros códigos pasados como también los que se realizaron para esta tarea en el repositorio.
- Logramos realizar con éxito los dos códigos solicitados usando el comando `system` que pertenece a la librería `stdlib.h`.
- Agregamos como colaborador al ingeniero a cargo del curso para que pueda verificar nuestros códigos y retroalimentarnos en el futuro.
- Aprendimos a subir nuestros códigos al repositorio, así como bajar las actualizaciones que pudieran hacer nuestros colaboradores.

<https://programavideojuegos.blogspot.com/2013/05/la-libreria-stdlibh.html>

## H. Referencias

<https://docs.github.com/es/get-started/start-your-journey/about-github-and-git>

[https://drive.google.com/file/d/1sZxzSUXsyC5eWiq\\_iLJQIFzPLQEnSYhC/view](https://drive.google.com/file/d/1sZxzSUXsyC5eWiq_iLJQIFzPLQEnSYhC/view)

[https://drive.google.com/file/d/1x3Cx1zu2Cxafpt\\_pxwbbazNRJVMt--MS/view](https://drive.google.com/file/d/1x3Cx1zu2Cxafpt_pxwbbazNRJVMt--MS/view)