Examen Final Blockchain

Asignatura:

Electiva Cpc

NRC:

21168

Docente:

Jaider Ospina Navas

.

Corporación Universitaria Minuto de Dios UNIMINUTO.

Ingeniería de sistemas.

Bogotá D.C., noviembre de 2021.

Manual y video implementación blockchain

Integrantes:

Cristian David Acosta Ramos

Ivan Dario Buitrago

Daniel David Enrique Guzmán Pinto.

David Sandoval Morales

David Garzón Mahecha

Sergio villabona chinchilla

Corporación Universitaria Minuto de Dios UNIMINUTO.

Bogotá D.C., agosto de 2021.

Tabla de contenido

Página

Introducción

El uso de blockchain demuestra que es una forma de desarrollo tecnológico muy importante por lo cual debemos abarcar muchos temas, con este concepto entendemos que desarrollar de una manera visual y escrita un manual, el cual nos explicara cómo es su empleo y desarrollo será útil a personas que no saben desarrollar en esta plataforma, podremos resaltar que blockchain es un sistema seguro, inmutable y confiable por lo cual muchas personas quieren aprender, en este manual podemos generar contrato inteligente el cual va enfocado hacia la súper intendencia de notariado y registro que se encarga de la emisión de los certificados de libertad y tradición, con la aplicabilidad de este contrato inteligente podremos evitar la modificación de la historia del inmueble dando así una certeza de no tener ventas ilícitas del mismo y demostrando la importancia e eficacia de la aplicabilidad de blockchain en la vida real.

Justificación del manual

- Facilitar la capacitación sobre las herramientas y el contenido de blockchain manejando un proceso claro y amigable para el usuario del manual.
- Demostrar conocimientos y manejo de las herramientas.
- Evidenciar como el manual puede ser útil en la vida real y en casos cotidianos.
- Generar una documentación y un paso a paso del desarrollo del contrato

Objetivos.

Objetivo general.

Desarrollar una solución para el manejo de blockchain que se requiere un manual el cual nos da acceso a información importante para lograr una mejor administración, control sobre un contrato inteligente para la súper intendencia de notariado y registro que se encarga de la emisión de los certificados de libertad y tradición, con la aplicabilidad de este contrato inteligente podremos evitar la modificación de la historia del inmueble

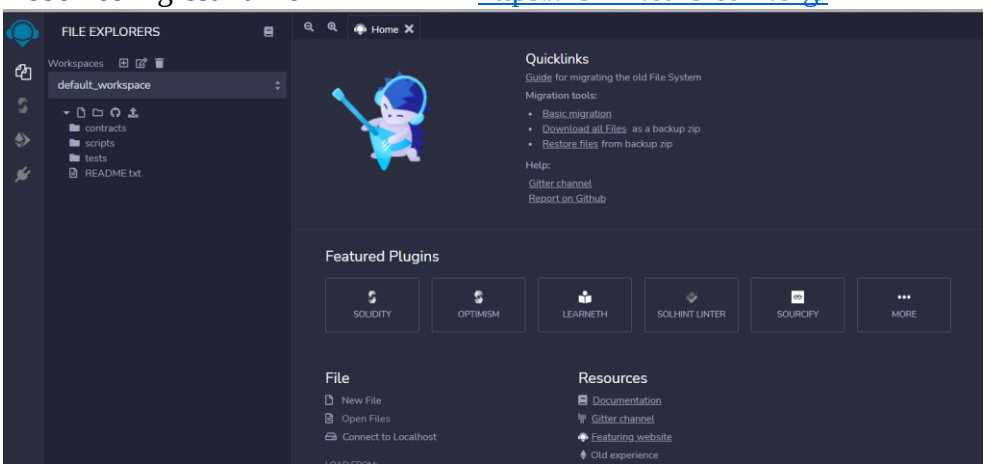
Objetivos específicos.

Generar tareas en el contrato inteligente

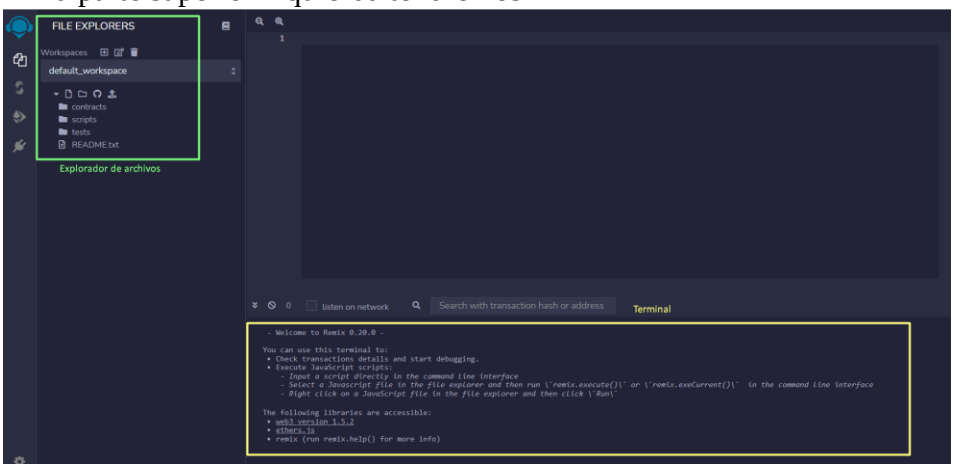
- Generar y guardar la información
- Eliminar o corregir información por el creador del contrato
- Seguimiento de la información

Manual

1. Debemos ingresar a Remix IDE link https://remix.ethereum.org/



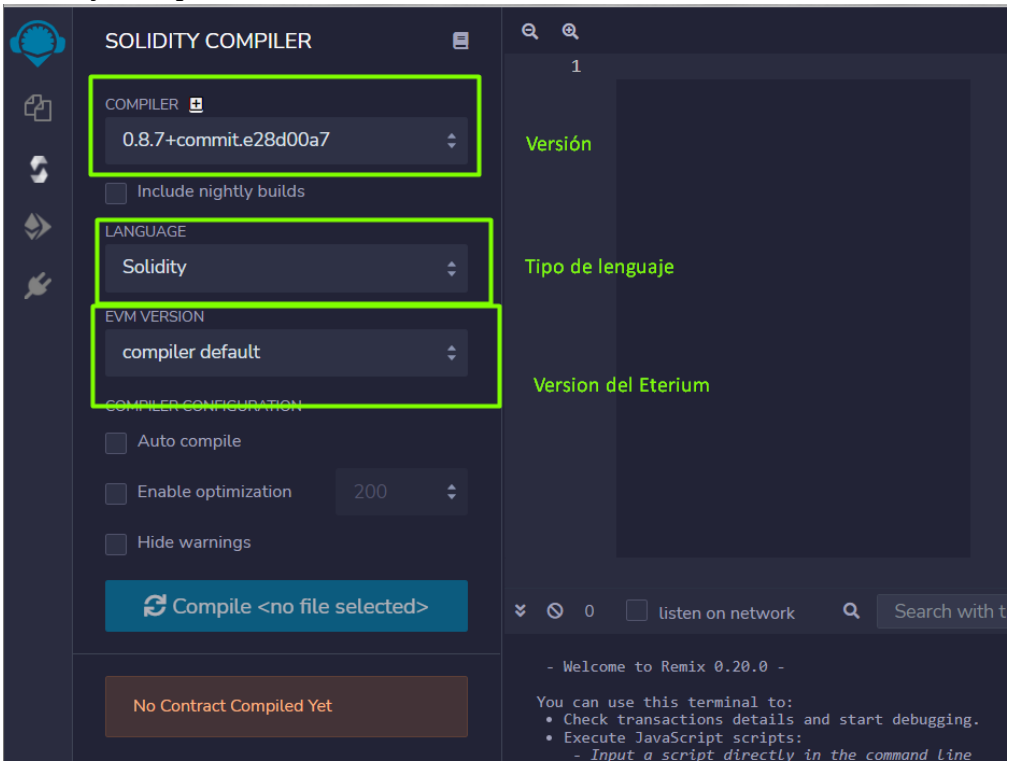
2. En la parte superior izquierda tendremos



El explorador de archivos nos permite visualizar y navegar a través de nuestro archivos, contratos, scripts y test.

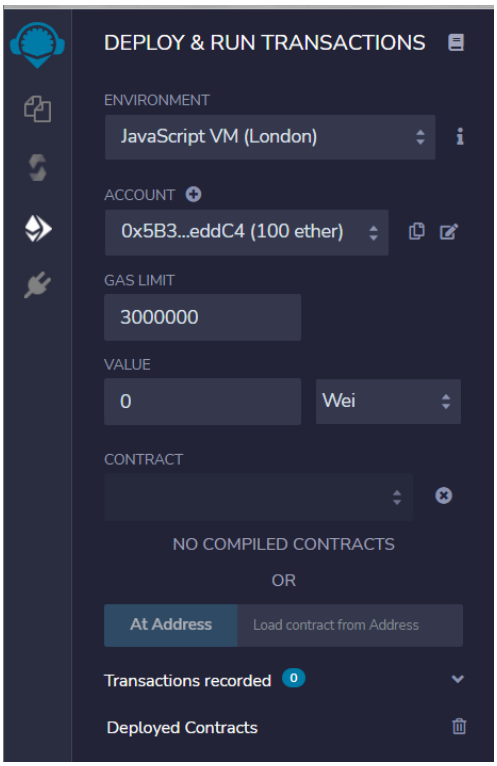
La terminal permite ver la ejecución de las funciones y los resultados de las mismas. En contratos es importante permite ver si las transacciones son exitosas o no.

3. Solidify Compiler



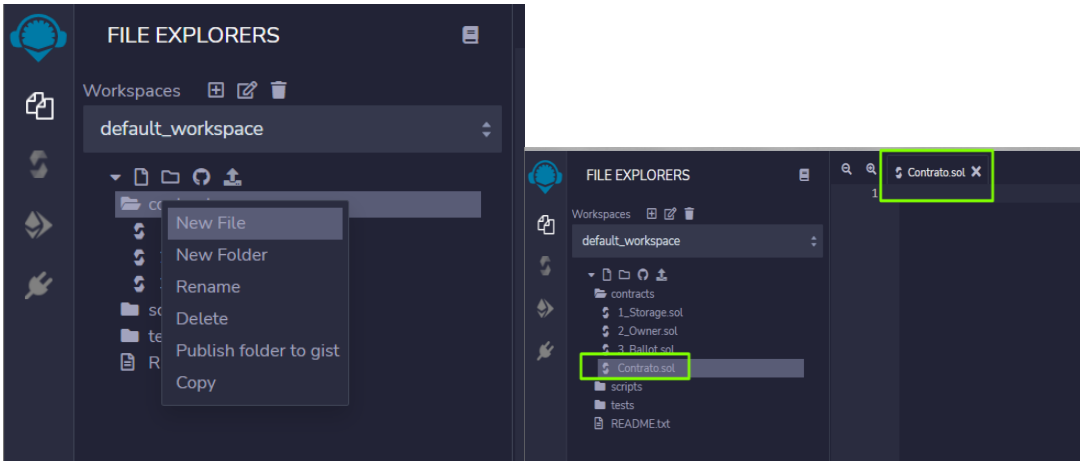
Es el compilador integrado del IDE, podemos seleccionar las versiones del lenguaje, el lenguaje que queremos usar y la versión del Eterium que nos va indicar donde se estará ejecutando el programa.

4. Transacciones



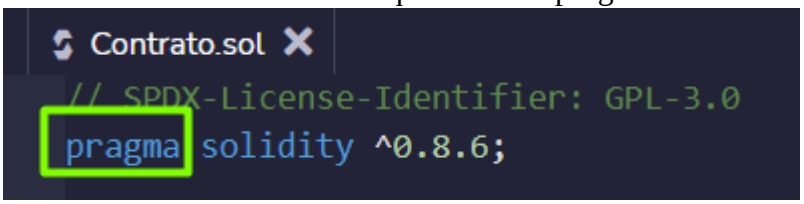
Permite que podemos subir los archivos compilados a Blockchain como subirlo a la nube.

5. Crear nuevo contrato



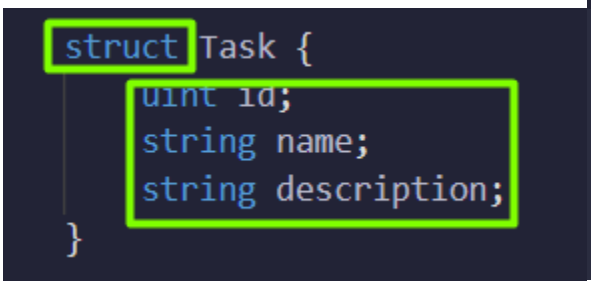
Recordar que cada contrato tiene su propio archivo y no es obligatorio que interactúe con otros.

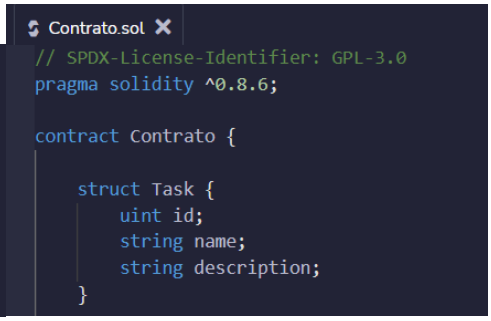
6. Se selecciona la versión en la que vamos a programar



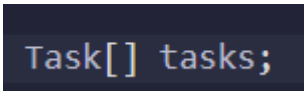
7. Al igual que otros lenguajes debemos instanciar la clase

8. Se procede a declarar variables

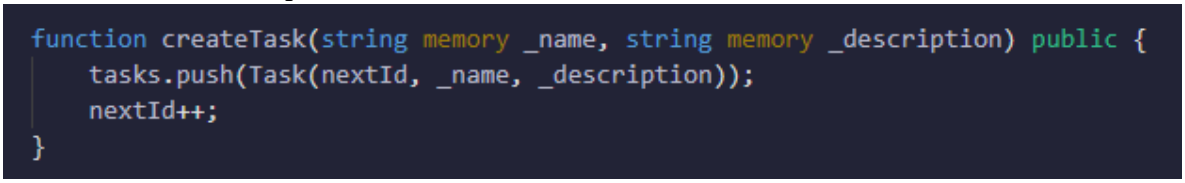




9. Creamos una lista de tareas y le asignamos un nombre



10. Creamos una función para las tareas



En este caso se crea una función que nos permita añadir datos a nuestra lista de tareas y guardarlas cuando se ejecute la función.

Usamos Next ID como variable de incremento esto nos permie que los ID se agreguen solos a medida que se van haciendo las tareas y el auto incrementable les dará el valor empezara en 0 e ira sumando 1 con su creación.

11. Se crea una función para traer datos de la tarea

```
function findIndex(uint _id) internal view returns (uint) {
    for (uint i = 0; i < tasks.length; i++) {
        if (tasks[i].id == _id) {
            return i;
        }
    }
    revert("Tarea no encontrada");
}
```

Se crea un ciclo for que recorra todas las tareas que tenemos
EL if se implementa para que nos traiga los datos de la transacción que nosotros digitemos
Llegado caso no encuentre el ID no retorna el mensaje que tenemos en “revert”.
Adicional cuando usamos “internal view” quiere decir que esta función no será visible es decir que por consola no nos va mostrar alguna información esta función será usada por otra.

12. Creación de función para leer nuestra lista de tareas y guardarla en el index

```
function readTask(uint _id) public view returns (uint, string memory, string memory) {
    uint index = findIndex(_id);
    return (tasks[index].id, tasks[index].name, tasks[index].description);
}
```

En a la función usamos el comando findIndex para el ID, nos retorna una tarea y guardarlas cuando se ejecute la función.

13. Actualizar tarea

```
function updateTask(uint _id, string memory _name, string memory _description) public {
    uint index =  findIndex(_id);
    tasks[index].name = _name;
    tasks[index].description = _description;
}
```

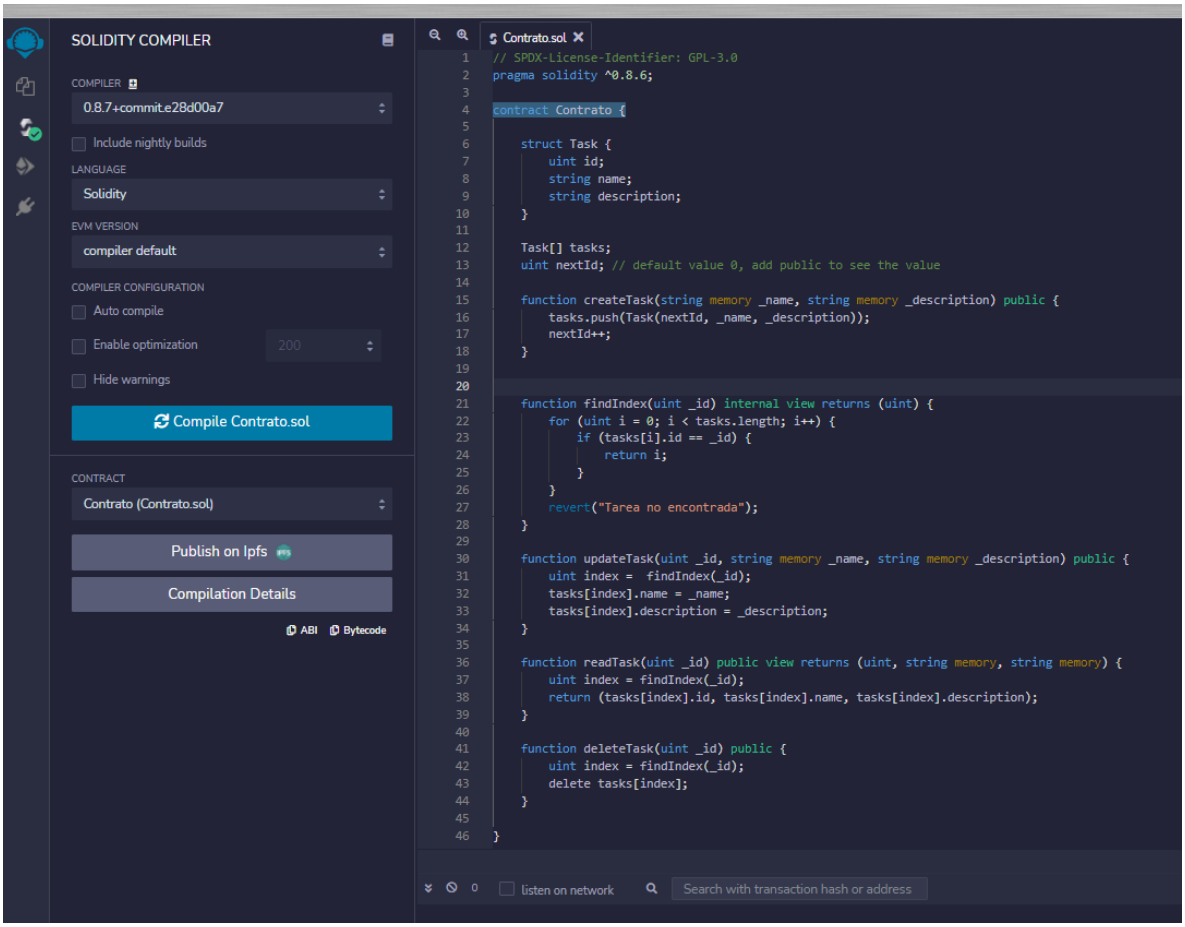
Esta tarea nos permite editar los datos de una tarea, solicitamos los nuevos datos y se actualizan.

14. Borrar tarea

```
function deleteTask(uint _id) public {
    uint index = findIndex(_id);
    delete tasks[index];
}
```

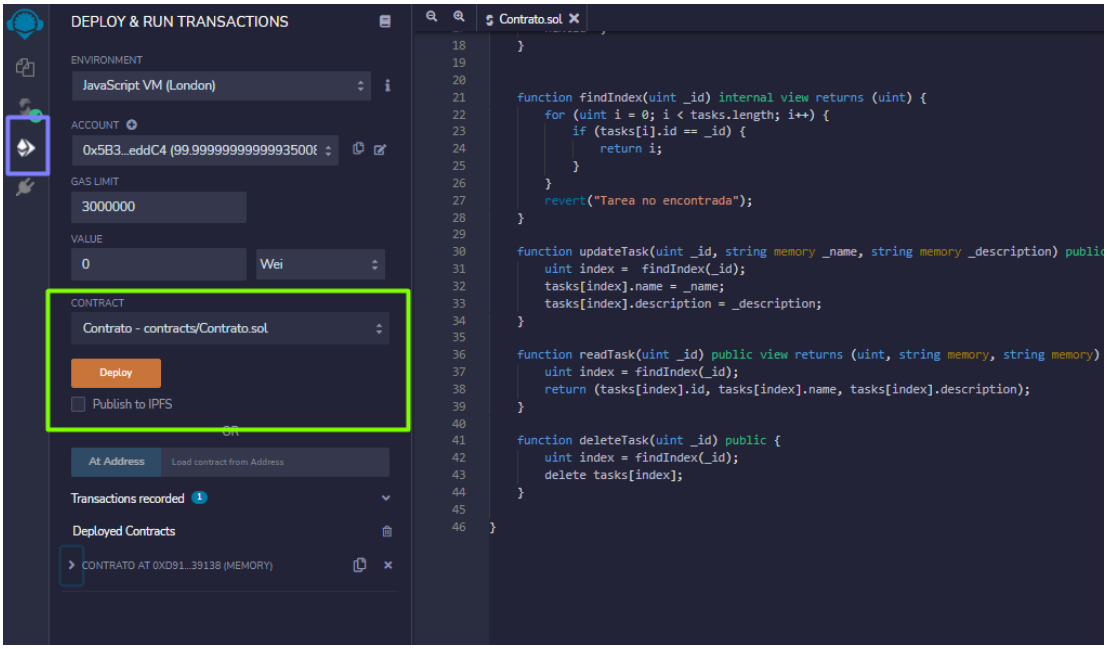
Esta tarea va buscar el ID ingresado y procederá a borrarla si se encuentra el ID.
Cabe mencionar que borrar un arreglo es algo complicado, pero para aplicarlo a BLOCKchain y que los datos sean históricos, lo que va hacer la función es a regresar los valores por defecto que tenia esa ubicación del array.

15. Ejecución del programa

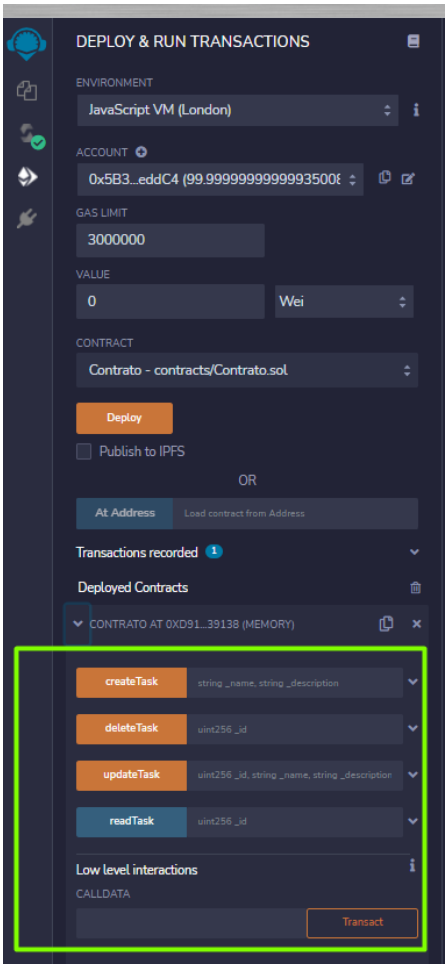


Se compila para validar que no haya ningún error.

Nos dirigimos a la tercera pestaña **Deploy and RUN** y se presiona **Display** para poderlo ejecutar.



Una vez ejecutado podremos utilizarlo y nos saldrán las opciones que habíamos programado.



Conclusiones

Encontramos que blockchain se ha convertido en una de las tecnologías que más interés suscita a nivel global. Por lo cual entendemos la importancia del mismo por lo cual hacemos este manual intentando construir soluciones, haciendo uso de estas cadenas de bloques., aunque ingeniosa, no es excesivamente compleja, sí lo es el construir una solución a gran escala donde cada participante entienda nuestro manual demostrando no solo los desafíos tecnológicos, sino que la dificultad también reside en juntar a todos los actores bajo un mismo consenso y unas mismas reglas. En este documento se ha tratado de explicar el funcionamiento de contrato inteligente, resaltando las características que la convierten en una tecnología diferente y que es usada en la vida real como con nuestro tema en la registraduria notarial. También hemos querido, sin embargo, dejar claro que blockchain no es una solución sino una herramienta y que, consecuentemente podrá ser usada sabiendo manejarla con un manual amigable al usuario y siguiendo el procedimiento impartido en el, también entendemos que debemos de tener ganas de seguir aprendiendo ya que tiene muchas formas de usar y es muy basto su uso.