**Manual de usuario de Numerical AS**

**Aspectos generales**

La aplicación fue diseñada en inglés para acercarse al público general, el nombre nace de los métodos de análisis numérico y la S de “sespinosav” el creador. El usuario debe tener en cuenta que la aplicación está escrita sobre Python y que el uso de los datos se trata con este lenguaje, esto es importante ya que en el momento de ingresar un número tipo flotante debido a esto es necesario que la “coma decimal” sea un punto, también en el caso de las funciones cuyas mismas se trabajan con la librería *math* de Python.

**El *header***



El header cuenta con una barra de navegación que incluye:

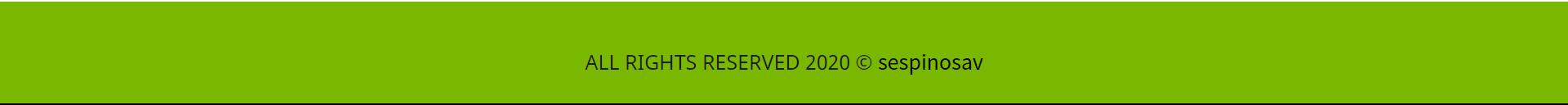
GUIE: un manual de usuario desde la web.

NUMERICAL METHODS: el apartado para utilizar todos los métodos numéricos.

CONVERTER: una aplicación para transformar un número incluyendo números decimales de una base a otra dentro del rango de base 2 a base 10.

MAKE MATRIX: una aplicación para la ayuda de creación de matrices en formato python.

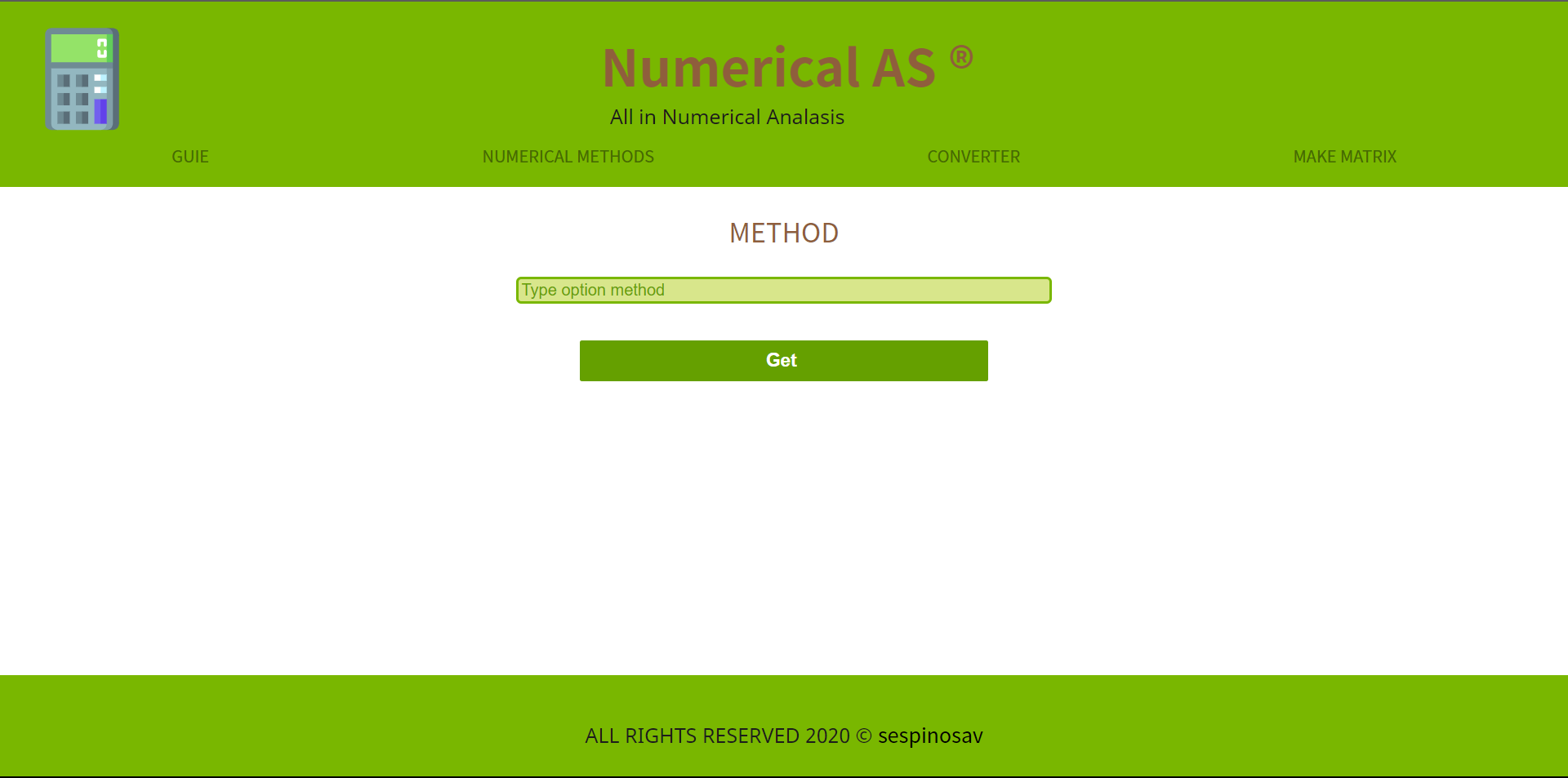
**El  *footer***



La parte inferior de la aplicación, que cuenta con los derechos de autor ya que la aplicación es totalmente original y un enlace para visitar el GitHub del creador.

El *footer* aparece o desaparece en cuestión del espacio que ocupen las vistas a mostrar, cosa que no sucede con el *header* que siempre está fijo.

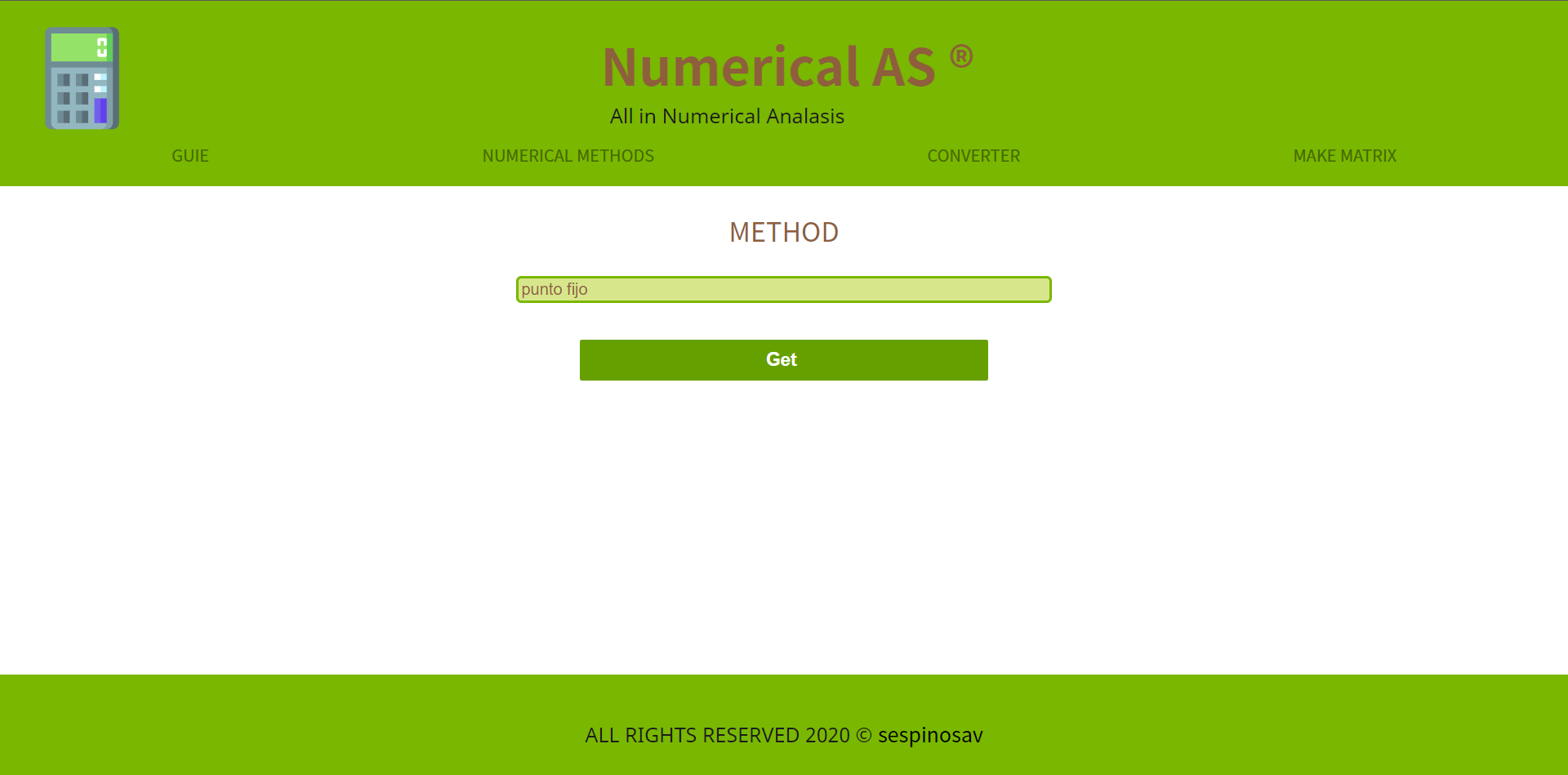
**NUMERICAL METHODS**



Está es la parte principal del programa y tiene tres vistas diferentes, la primera que se acaba de mostrar en donde se escribe el método numérico a emplear, se puede escoger entre los siguientes:

* busquedas incrementales
* biseccion
* regla falsa
* newton
* punto fijo
* secante
* raices multiples
* eliminacion gaussiana simple
* eliminacion gaussiana con pivoteo parcial
* eliminacion gaussiana con pivoteo total
* lu simple
* lu parcial
* crout
* doolittle
* cholesky
* jacobi
* gauss seidel
* sor
* vandermonde
* interpolante de newton sin diferencias divididas
* interpolante de newton
* lagrange
* trazador lineal
* trazador cuadratico
* trazador cubico

Es importante que ningún método ingresado no tenga tildes ni espacios de más.



Ejecutaremos un ejemplo con el método de punto fijo, a la hora de ingresar el método no importan las mayúsculas o minúsculas, luego de esto se puede dar en el botón *Get* o dando enter.

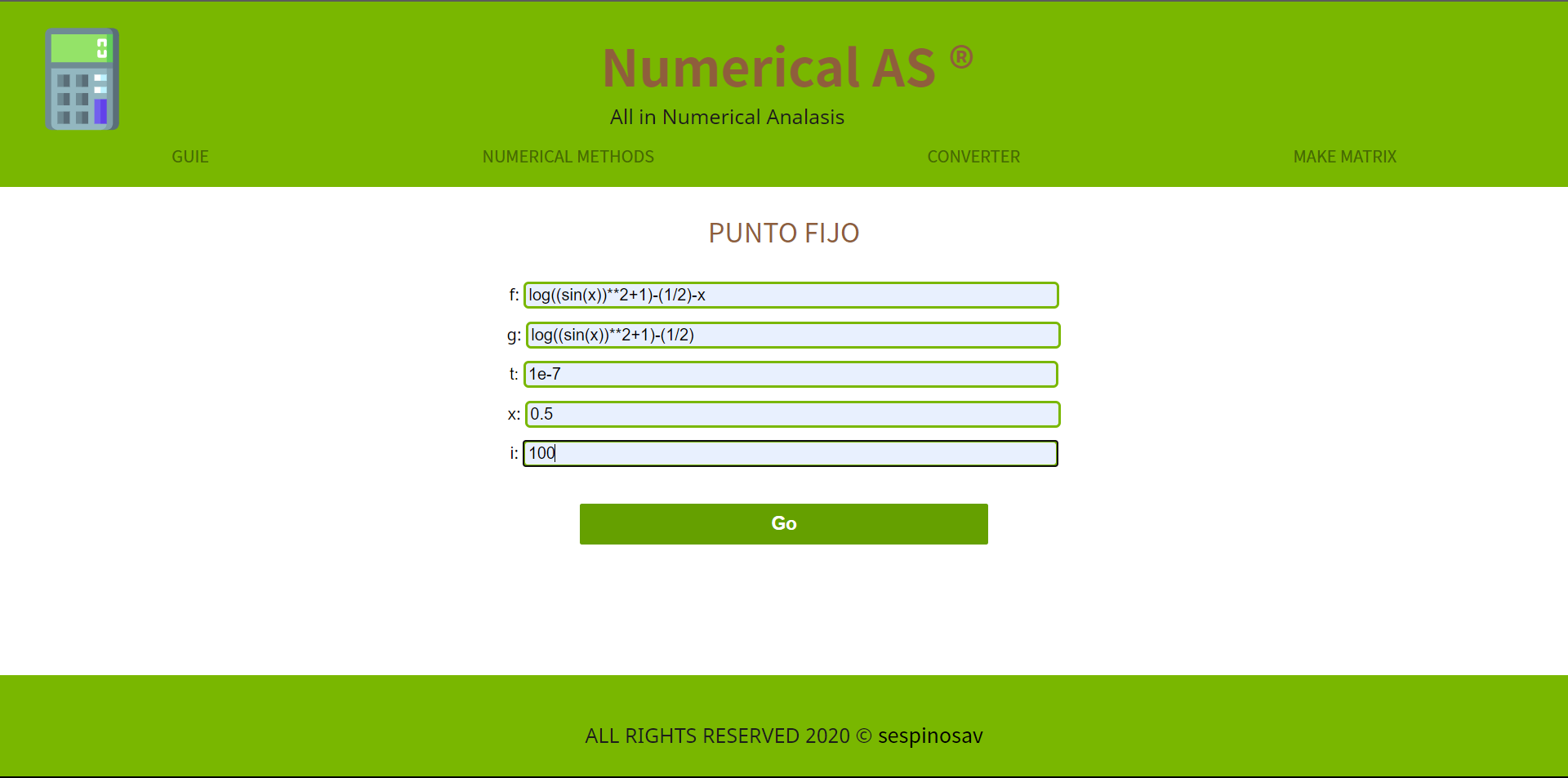


Luego aparecerá un formulario con los diferentes campos necesarios para el método, en el caso de los métodos que necesitan funciones, estás pueden ser ingresadas de dos maneras.

Utilizando los módulos por defecto de math, cómo cos(), sin(), log() y de más o desde math directamente con math.cos(), math.sin() y con todas las funciones que soporte la librería, para saber todas las funciones que soporta math, puedes revisar el siguiente enlace: <https://www.programiz.com/python-programming/modules/math>

Para este ejemplo usaremos los siguientes datos:

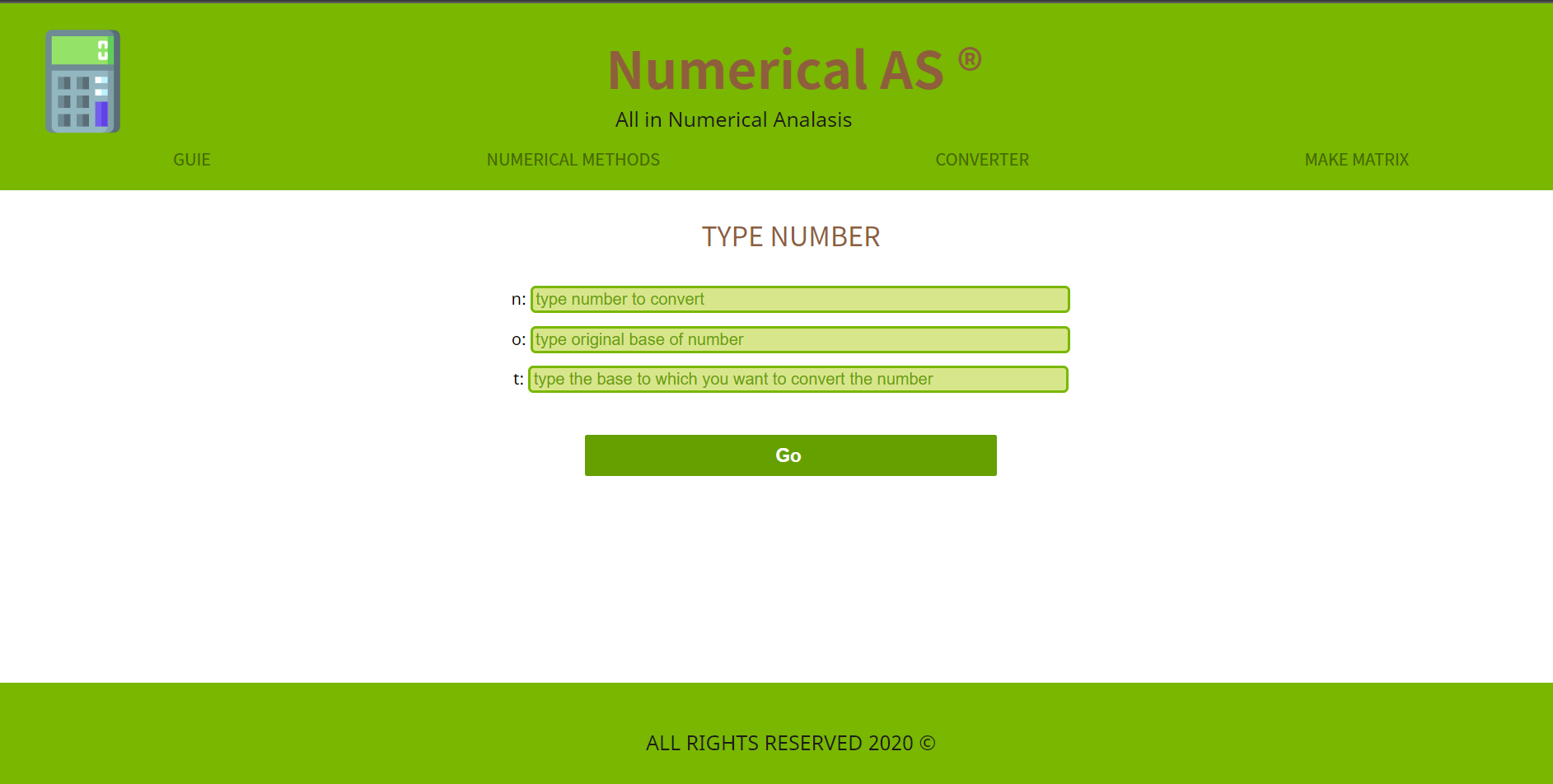
* log((sin(x))\*\*2+1)-(1/2)-x
* log((sin(x))\*\*2+1)-(1/2)
* 1e-7
* 0.5
* 100



Luego le damos a *enter* o a el botón *Go.*



Nos aparecerá el resultado para estos datos y un botón para regresar, este botón nos devuelve al menú de Numerical Methods, si deseamos ingresar otra serie de datos para el mismo método, debemos darle atrás a la pestaña, o en el botón y volver a ingresar el método.

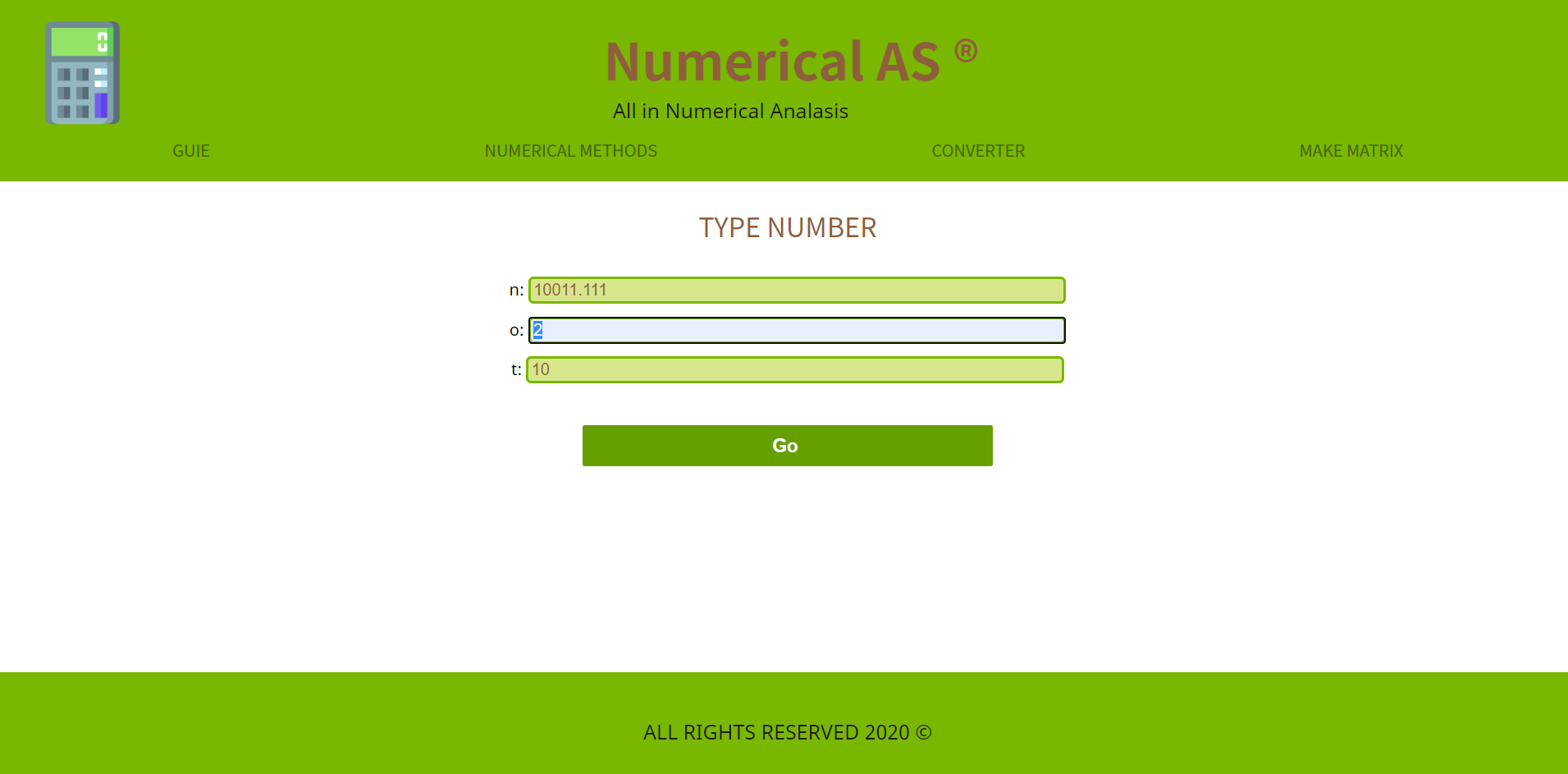


La ventana *converter* nos deja transformar un número de una base entre 2 a 10 a otro número en el mismo rango de bases (este programa soporta números flotantes), con el mismo funcionamiento del Numerical Methods.

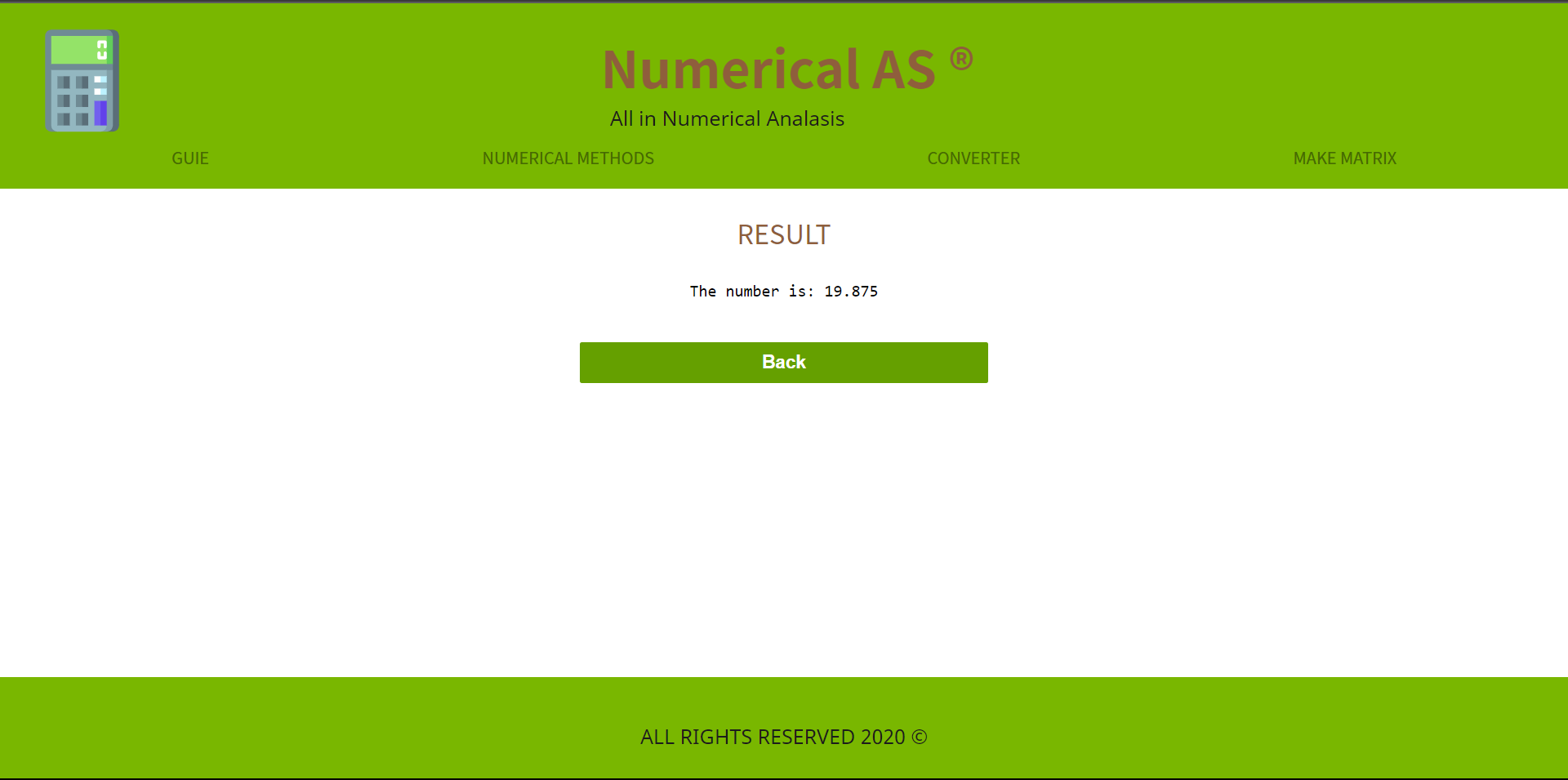
Ingresamos los datos, para un ejemplo, usaremos estos:

* 10011.111
* 2
* 10

El primer campo es el número que deseamos convertir, el segundo la base original y el tercero la base buscada.

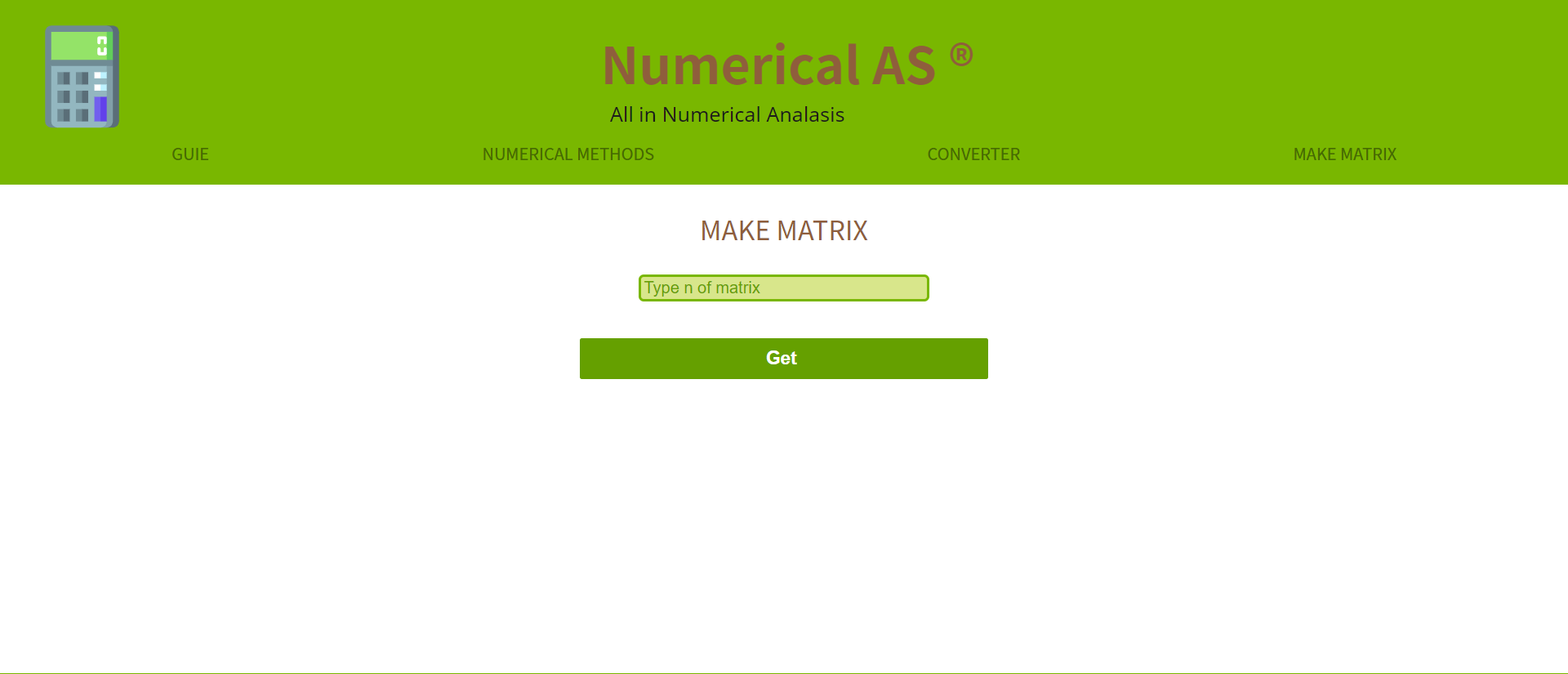


Le damos a *Go* y se nos muestra el resultado.

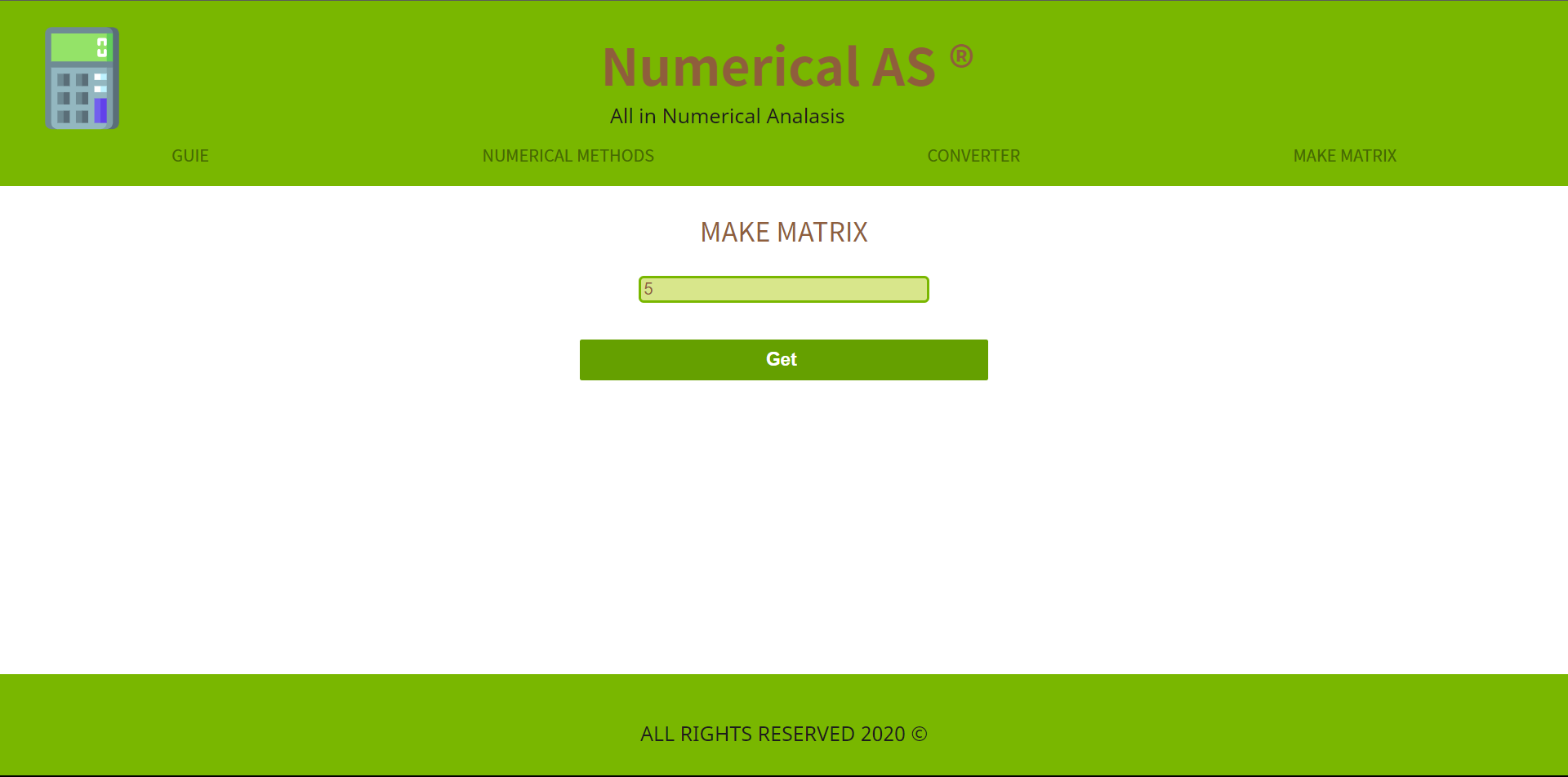


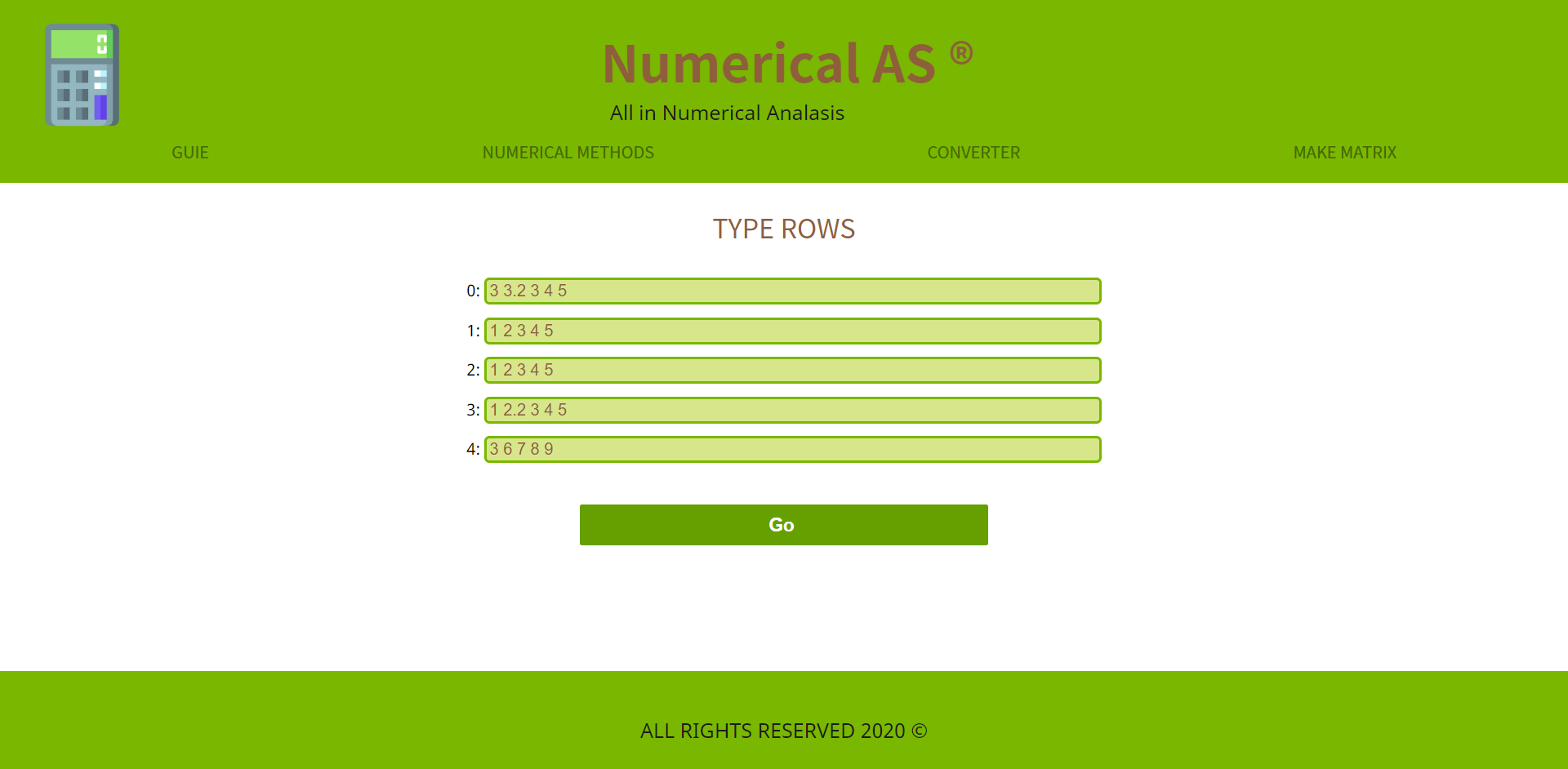
El botón *Back* nos regresa al menú de *CONVERTER.*

**MAKE MATRIX**

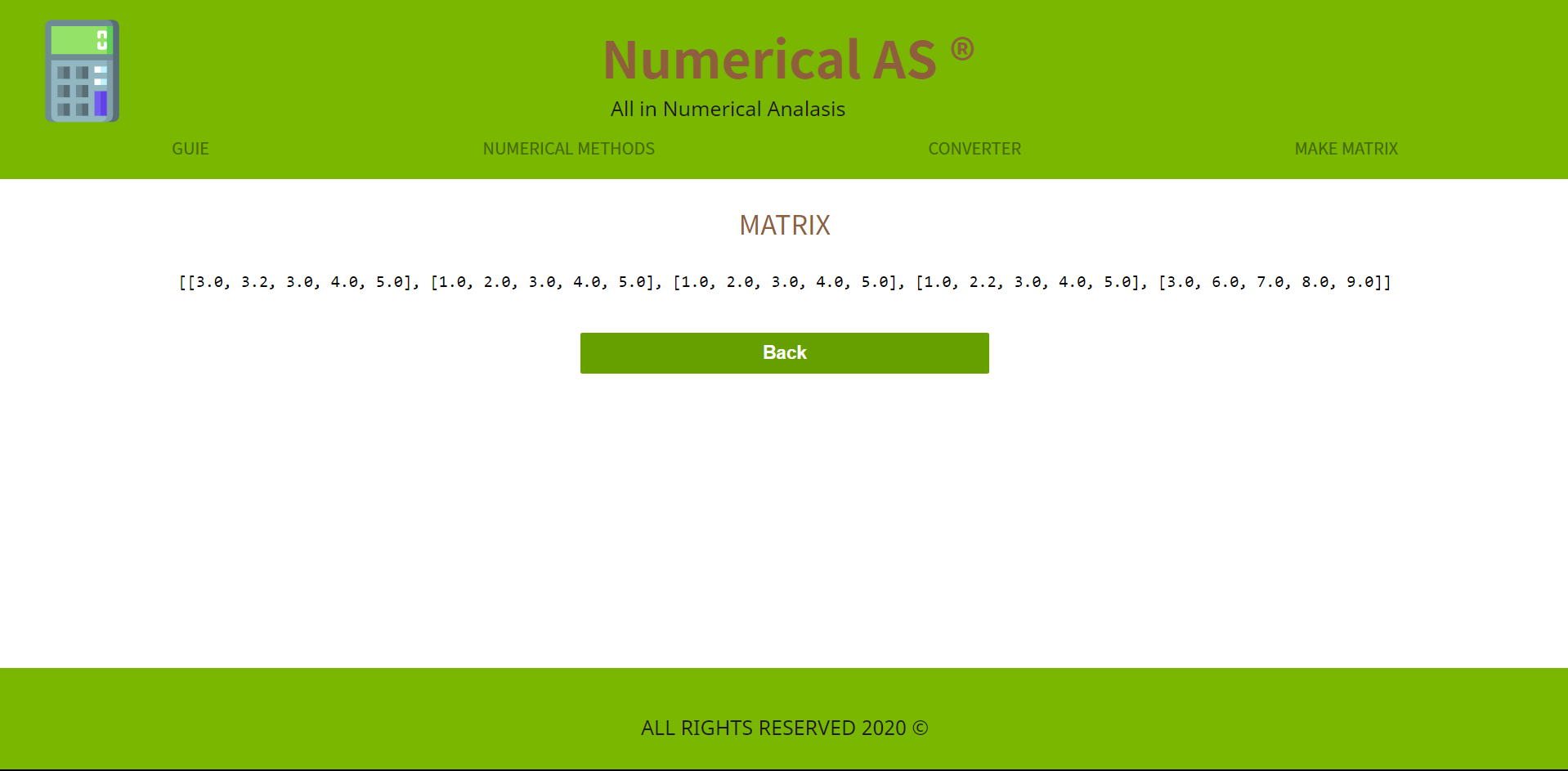


Para los métodos numéricos que necesitan matrices o arreglos, cómo ‘eliminación gaussiana simple’ podemos usar *MAKE MATRIX* que nos ayuda a construir una matriz con n filas en formato pytho, para este ejemplo usaremos una matriz con 5 filas.

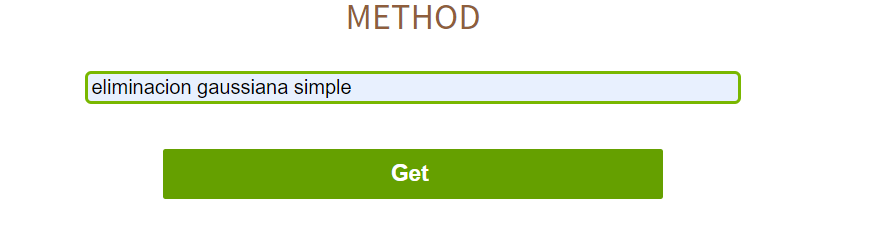


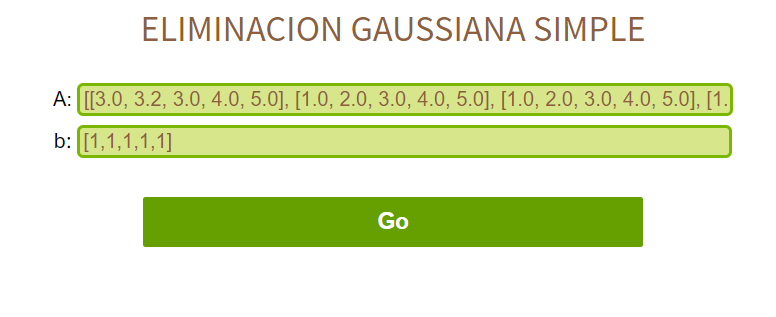


Los datos ingresados deben estar separados por espacios. Se empieza desde la fila 0 ya que esa es la convención en python. Luego del resultado, obtenemos:

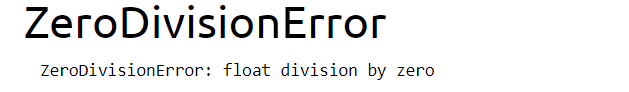


Podemos tomar esta matriz para usar un método que necesite de estás, hagamos un ejemplo con el método de ‘eliminación gaussiana simple’.





Usaremos el arreglo [1,1,1,1,1] para el arreglo b.



Si conocemos sobre el tema sabremos que esto indica que el sistema no tiene solución única.

**Conclusiones**

La aplicación es intuitiva pero hay que tener cuidado a la hora de ingresar los datos ya que no se cuenta con una barra de opciones para los métodos y se usa un formato de matrices que puede ser confuso, pero en los otros aspectos no debería generar muchas dificultades.