

## UNIVERSIDAD POLITÉCNICA DE PÉNJAMO(UPP)



**Nombre del alumno:** Brandon Iván Márquez Morales.

**Grado y Grupo:** 1° "A"

**Carrera:** Ingeniería en Software.

**Materia:** Algoritmos.

**Tema:** Practica con arreglos bidimensionales (Gato o del Tic Tac Toe)

**Fecha de inicio:** 16/11/2020.

**Fecha de término:** 19/11/2020.

**Docente:** Miguel Ángel Saldaña Cabeza.

## Contenido

Introducción. ....	3
Marco teórico. ....	4
¿Qué es una variable global? .....	4
Diferencia entre variable local y global. ....	4
Asociatividad. ....	4
Existencia. ....	4
Confiabilidad. ....	4
¿Qué es una función? .....	5
¿Qué es una función con parámetros? .....	5
Múltiples parámetros. ....	5
¿Qué es una función con retorno? .....	5
¿Qué es un arreglo? .....	6
¿Qué es un arreglo vectorial? .....	6
¿Qué es un arreglo matricial? .....	7
Arreglos y cadena de caracteres .....	7
¿Qué es una excepción? .....	8
Desarrollo. ....	9
1.- Se crea arreglo matricial de 3x3. ....	9
2.- Se crea la función para mostrar el tablero. ....	9
3.- Se crea la función para limpiar el tablero. ....	9
4.- Se crea función para validar la lectura de las posiciones. ....	10
5.- Se crea la función para el turno de "X" y "O". ....	10
6.- Se crea un ciclo para estar repitiendo el juego si es que así se desea. ....	11
Conclusiones. ....	14
Bibliografía .....	15

## Introducción.

En el presente documento se dará a conocer el desarrollo de un algoritmo que sobre un juego muy conocido “El gato” o “Tic Tac Toe” desarrollo en los tres lenguajes de programación diferentes que se vienen manejando en este cuatrimestre (C++, Python y Ruby).

En los últimos temas nuevos que se han visto de algoritmos son de funciones con y sin parámetros y retorno y arreglos vectoriales y matriciales, estas nuevas herramientas de programación adquiridas nos ayudado principalmente a reducir nuestro código y tener un mejor entendimiento a nuestros códigos de programación.

Viendo estas nuevas herramientas se han desarrollado algunas prácticas para dominar las funciones y arreglos, es por ello que ahora se nos asignó la práctica del “Gato” el cual se explicará como se fue desarrollando paso a paso y explicando para que sirve cada herramienta en el código.

Además de retomar información de cada una de las cosas nuevas vistas del tema en el marco teórico que se mostrará a continuación.

## Marco teórico.

### ¿Qué es una variable global?

Una variable global es una variable que se declara fuera de todas las funciones dentro del programa. Puede haber múltiples funciones en un solo programa. Todas estas funciones pueden acceder a estas variables globales. Por lo tanto, una variable global no es muy segura ya que el valor puede ser cambiado por otras funciones. Existe una variable global hasta completar la ejecución de todo el programa. (STREPHONSAYS, 2020)

Las variables globales son útiles cuando múltiples funciones operan en los mismos datos. Por otro lado, el valor de una variable global no es confiable, ya que se puede cambiar. (STREPHONSAYS, 2020)

### Diferencia entre variable local y global.

La variable local es una variable que se declara dentro de una función de un programa de computadora. La variable global es una variable que se declara fuera de las funciones de un programa de computadora. Esta es la diferencia básica entre la variable local y global. (STREPHONSAYS, 2020)

### Asociatividad.

Además, mientras que una variable local es accesible solo dentro de la función que se declara, una variable global es accesible por todas las funciones en el programa. (STREPHONSAYS, 2020)

### Existencia.

Otra diferencia importante entre la variable local y global es su existencia. Existe una variable local hasta que la función se ejecuta. La variable local se crea cuando la función comienza a ejecutarse y se destruye cuando se completa la ejecución de la función. Por otro lado, una variable global permanece en existencia durante todo el tiempo que el programa se está ejecutando. (STREPHONSAYS, 2020)

### Confiabilidad.

Además, una variable local es más confiable y segura, ya que otras funciones no pueden cambiar el valor. Por el contrario, una variable global es accesible por múltiples funciones. Por lo tanto, su valor puede ser cambiado. (STREPHONSAYS, 2020)

### ¿Qué es una función?

Una función es un grupo de instrucciones con un objetivo en particular y que se ejecuta al ser llamada desde otra función o procedimiento. Una función puede llamarse múltiples veces e incluso llamarse a sí misma (función recursiva). (ALEGSA.com.ar, 2020)

### ¿Qué es una función con parámetros?

Los parámetros se usan para mandar valores a las funciones. Una función trabajará con los parámetros para realizar las acciones. Por decirlo de otra manera, los parámetros son los valores de entrada que recibe una función. (desarrolloweb.com, 2020)

Por poner un ejemplo sencillo de entender, una función que realizase una suma de dos números tendría como parámetros a esos dos números. Los dos números son la entrada, así como la salida sería el resultado de la suma. (desarrolloweb.com, 2020)

### Múltiples parámetros.

Una función puede recibir tantos parámetros como queramos y para expresarlo se colocan los nombres de los parámetros separados por comas, dentro de los paréntesis. (desarrolloweb.com, 2020)

### ¿Qué es una función con retorno?

Una función con retorno es un fragmento de código diseñado con el objetivo de ser lo más reutilizable posible y que, tras ejecutarse, devuelve un valor numérico, textual, lógico o de otros tipos que será utilizado por el programa principal. Las funciones con retorno son ampliamente usadas ya que permiten, junto con el uso de argumentos, crear fragmentos de código que pueden ser usados en múltiples programas sin necesidad de reescribirlos en cada situación.

### ¿Qué es un arreglo?

Un arreglo es una estructura, es decir es un conjunto de variables que se citan y manejan con un mismo nombre, y que permite además la utilización individual de sus elementos. (Programaciónestructurada, 2018)

Esto significa que un arreglo es una variable que provee la opción de disponer de varios datos o espacio para datos a los cuales se puede acceder con un mismo nombre. (Programaciónestructurada, 2018)

Un arreglo es en resumen un conjunto de datos finito y del mismo tipo. En realidad, funciona como cualquier variable cualquiera, excepto que, en lugar de almacenar un solo valor, guarda algunos valores. (Programaciónestructurada, 2018)

### ¿Qué es un arreglo vectorial?

Es un arreglo de “N” elementos organizados en una dimensión donde “N” recibe el nombre de longitud o tamaño del vector. Para hacer referencia a un elemento del vector se usa el nombre del mismo, seguido del índice (entre corchetes), el cual indica una posición en particular del vector. (pdf, 2019)

Por ejemplo: V[x]

Donde:

V Nombre del arreglo

x Número de datos que constituyen el arreglo(tamaño)



## Representación gráfica



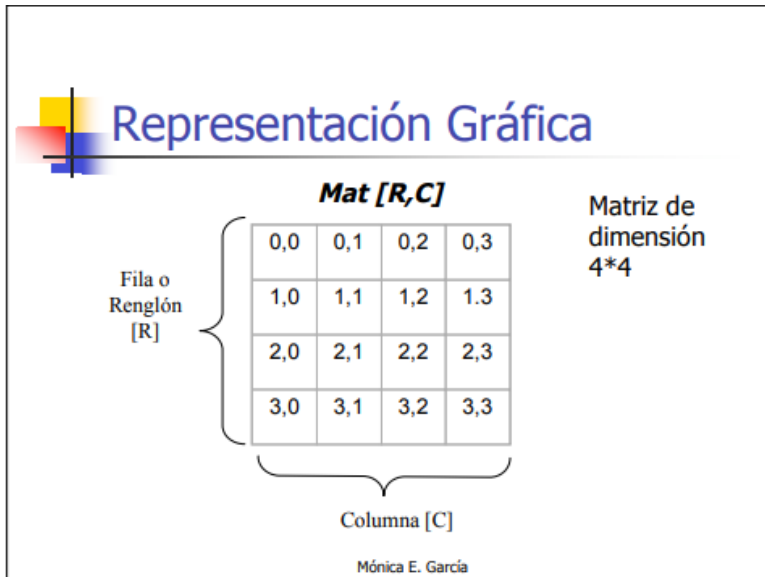
V[0]	7
V[1]	8
V[2]	9
V[3]	10

**Nota sobre los arreglos**  
El valor del índice debe de estar entre 0 y un LIMITE entero

Mónica E. García

### ¿Qué es un arreglo matricial?

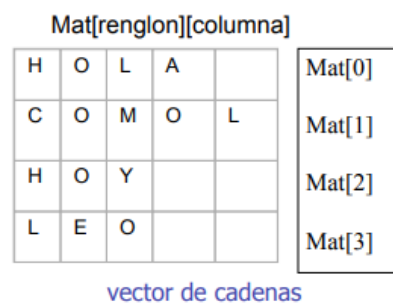
Es un arreglo de  $M * N$  elementos organizados en dos dimensiones donde  $M$  es el número de filas o renglones y  $N$  el número de columnas. Para representar una matriz se necesita un nombre de matriz acompañado de dos índices.  $Mat [R,C]$  Donde  $R$  indica el renglón y  $C$  indica la columna, donde se encuentra almacenado el dato. (pdf, 2019)



### Arreglos y cadena de caracteres

Una cadena de caracteres se representa por un arreglo unidimensional de caracteres. Cada carácter de la cadena se almacena en un elemento del arreglo. En general el último elemento de la cadena es el carácter nulo ('\0'). (pdf, 2019)

### Representación gráfica



### ¿Qué es una excepción?

Una excepción es la indicación de que se produjo un error en el programa. Las excepciones, como su nombre lo indica, se producen cuando la ejecución de un método no termina correctamente, sino que termina de manera excepcional como consecuencia de una situación no esperada. (powered, 2018)

Cuando se produce una situación anormal durante la ejecución de un programa (por ejemplo, se accede a un objeto que no ha sido inicializado o tratamos de acceder a una posición inválida en un vector), si no manejamos de manera adecuada el error que se produce, el programa va a terminar abruptamente su ejecución. Decimos que el programa deja de funcionar y es muy probable que el usuario que lo estaba utilizando ni siquiera sepa qué fue lo que pasó. (powered, 2018)



## Desarrollo.

### 1.- Se crea arreglo matricial de 3x3.

En este caso el tipo de arreglo que se debe crear es de tipo "string" ya que los símbolos del tablero es un carácter en este es un: "\_" y la dimensión se crea de 3x3 porque así debe de ser el tablero de nueve posiciones.

### 2.- Se crea la función para mostrar el tablero.

Se creó un ciclo for dentro de otro ciclo for el primer ciclo for es para que el iterador pase por cada una de las filas del arreglo del tablero y el segundo ciclo for es para que el iterador pase por cada una de las columnas del arreglo del tablero en cada ciclo for el iterador se inicia en cero y su limite es de menor o igual dos entonces tenemos con esto ya sabemos que nuestro iterador recorrerá 3 posiciones siempre y cuando seguidamente le indiquemos que el iterador ira incrementando.

Después del segundo ciclo for se imprime el arreglo del tablero indicando las posiciones del iterador entre corchetes [].

### 3.-Se crea la función para limpiar el tablero.

Primero se pone el nombre el arreglo del del tablero que es "tablero" después en un primer corchete[] se pone la posición de la fila y en segundo corchete[] se pone la posición de la columna y así sucesivamente para cada posición y finalmente se le asigna el valor original que se le asignó desde un inicio al crear el arreglo("\_"). Esto nos servirá para que cuando haya un ganador o no halla un ganador nos pregunte si queremos jugar de nuevo y así el tablero no tenga los símbolos "X" y "O".

#### 4.-Se crea función para validar la lectura de las posiciones.

En caso de C++ en la función se hace una conversión de cadena a entero necesitamos 2 valores uno de tipo entero y otro de tipo string, todo esto se maneja dentro de un ciclo con una bandera de tipo bool iniciada en true entonces lee el primer valor que es de tipo string después el valor2 va a ser igual a la conversión del valor1 y automáticamente se le asigna un valor en 0 después se compara si no se ingreso cadena y si no bandera = false y ya no se repite, caso contrario se manda mensaje de “solo se admiten números” y bandera se asigna ha true para volver a ingresar la posición del tablero.

En caso de Ruby se hizo exactamente lo mismo “una conversión” pero se interpreta diferente.

En caso de Python se utilizó “try exept” básicamente se usa una bandera como en el anterior caso para estar repitiendo cada que no se inserte un entero al pedir las posiciones, dentro del “try” se intenta leer las posiciones del tablero si se introdujo un entero se rompe la bandera si da una excepción se da mensaje “solo se admiten números” y después nos retornamos a la lectura de la posición con “return”

#### 5.- Se crea la función para el turno de “X” y “O”.

Básicamente es lo mismo por eso se pone junto, primero se pide la fila ya sea de “X” o “O” y a la variable de fila se le va asignar la validación anterior que se explicó, después de eso a esa variable se le resta uno (-1) para que nosotros como usuarios iniciemos desde 1, ya que en los arreglos se inicia desde 0, Después se pide la columna y es exactamente el mismo proceso que la fila.

Después se va a comparar si la fila o la columna son mayores o igual a 3, esto porque recordemos que nuestro tablero es de 3x3 si se dan unas dimensiones superiores se rompe el programa, entonces si se cumple esto se da mensaje: "NO PUEDE DAR VALORES MAYORES A 3!!, ELIGA ENTRE 1-3..." y posteriormente cumplimos la condición de nuestro ciclo para volver ha pedir fila y columna.

Caso contrario se compara si dicha fila y columna están vacías, es decir que sean igual a “\_” si eso se cumple entonces a nuestro arreglo “tablero” se le asigna el símbolo ya sea “X” o “O” según el turno que se este tratando y rompemos nuestro ciclo des cumpliendo la condición que pide el ciclo asignándole cualquier otro valor diferente del que se pide.

Caso contrario de que dicha fila y columna no estén vacías o que sean igual a “\_” significa que posición está ocupada y volvemos a pedir fila y columna cumpliendo la condición de nuestro ciclo.

6.- Se crea un ciclo para estar repitiendo el juego si es que así se desea.

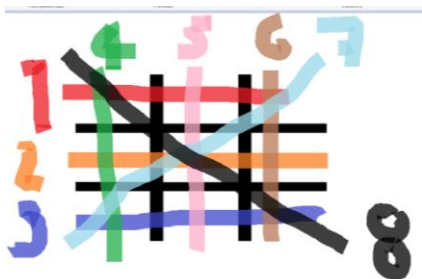
“Todo lo que se mencionará a continuación se estará repitiendo hasta que rompamos el ciclo”

1.-Limpiamos consola.

2.-Mostramos el tablero

3.-Llamamos la función del turno de “X” (que ya sabemos para que sirve)

4.-Se va a comparar si alguna de las 8 posiciones para poder ganar se igual a “X” para ello se dice: si fila1,columna1 es igual a “X” && fila1,columna2 es igual a “X” && fila1,columna3 es igual “X” seguidamente en esa comparación se utiliza el or(||) para seguir comparando las posiciones faltantes de poder ganar, sabemos que este es el número de posiciones para poder ganar:



Después si se cumple alguna posición se da un mensaje de que gano “X”, seguidamente se muestra el tablero actualizado para comprobar

que si ganó y se rompe o termina el juego, Luego se pregunta si se quiere volver a jugar, se lee la variable del ciclo y si se cumple se inicia de nuevo el juego, pero antes se llama a la función para limpiar el tablero.

Si no se cumplió alguna de las posiciones de "X" se va a comparar si cada una de las posiciones del tablero son diferentes de vacío("\_") esto para decir que no hay ningún ganador es decir que todas las posiciones ya están ocupadas ya sea por "X" o "O".

¿Por qué sigue comparar lo anterior?

Porque en la vuelta del ciclo #9 se piden las posiciones para "X" entonces, estando en la vuelta #9 y una vez darle posición a "X" y si no ganó ninguno entonces sigue comprar esto, En cambio si después de pedir posiciones para "X" y después pedimos posiciones para "O" siguiendo esta jerarquía que se menciona nunca saldría del ciclo ya que en la misma vuelta del ciclo #9 al dar la posición de "X" todas las casillas ya están ocupadas en caso de que ni "X" y "O" hallan ganado anteriormente y esa es la razón de este orden.

Ahora si se cumple esto (todas las posiciones  $j=$  de "\_"), entonces se da mensaje de que no hay ningún ganador y se muestra el tablero actualizado para comprobarlo. Luego se pregunta si se quiere volver a jugar, se lee la variable del ciclo y si se cumple se inicia de nuevo el juego, pero antes se llama a la función para limpiar el tablero.

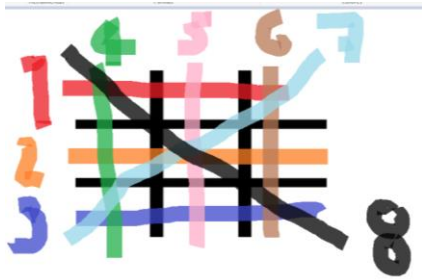
Caso contrario finalmente:

1.-Limpiamos consola.

2.-Mostramos el tablero

3.-Llamamos la función del turno de "O" (que ya sabemos para que sirve)

4.-Se va a comparar si alguna de las 8 posiciones para poder ganar se igual a "O" para ello se dice: si fila1,columna1 es igual a "O" && fila1,columna2 es igual a "O" && fila1,columna3 es igual "O" seguidamente en esa comparación se utiliza el or(||) para seguir comparando las posiciones faltantes de poder ganar, sabemos que este es el número de posiciones para poder ganar:



Después si se cumple alguna posición se da un mensaje de que gana "O", seguidamente se muestra el tablero actualizado para comprobar que si ganó y se rompe o termina el juego, Luego se pregunta si se quiere volver a jugar, se lee la variable del ciclo y si se cumple se inicia de nuevo el juego, pero antes se llama a la función para limpiar el tablero.

Y finalmente pues se estaría repitiendo nueve veces en caso de no ganar ni "X" ni "O" anteriormente.

## Conclusiones.

Con estas últimas prácticas vistas he aprendido bastante sobre los arreglos vectoriales y matriciales y funciones con y sin parámetros, además de algunas funciones que pueden tener un retorno o no.

Con esta última práctica de arreglos con funciones me he llevado bastante experiencia retomando las practicas anteriores como la de las recargas, en realidad en esta práctica del gato no tuve ningún tipo de problema para desarrollar el algoritmo me fue bastante fácil.

Además, tampoco tuve ningún problema con alguno de los lenguajes que se están manejando (C++, Python y Ruby), ya sea con alguna sintaxis o algo así ya que la verdad ya se ha practicado bastante los diferentes lenguajes con todas las prácticas que se han realizado desde que inicio el cuatrimestre y eso ha favorecido mucho el estar practicando desde un inicio.

Y finalmente para mí los arreglos ya sea vectoriales o matriciales son útiles para el almacenamiento de datos ya que igualmente se puede hacer con variables independientes, pero eso no es recomendable ya que se gastaría mucho espacio en memoria.

Y sobre las funciones de igual manera son muy útiles para reutilizar código de un mismo bloque, de esta manera pues nos podemos ahorrar varias líneas de código y así nuestros programas serán más entendibles ya que no serán tan extensos.

## Bibliografía

- ALEGSA.com.ar. (2020). *ALEGSA.com.ar*. Obtenido de ALEGSA.com.ar: <https://www.alegsa.com.ar/Dic/funcion.php#:~:text=Definici%C3%B3n%20de%20Funci%C3%B3n%20%28programaci%C3%B3n%29%20En%20programaci%C3%B3n%2C%20una%20funci%C3%B3n,e%20incluso%20llamarse%20a%20s%C3%AD%20misma%20%28funci%C3%B3n%20recursiva%29>.
- desarrolloweb.com. (2020). *desarrolloweb.com*. Obtenido de desarrolloweb.com: <https://desarrolloweb.com/articulos/585.php>
- pdf. (2019). *pdf*. Obtenido de pdf: [http://www.utm.mx/~mgarcia/PE4\(Arreglos\).pdf](http://www.utm.mx/~mgarcia/PE4(Arreglos).pdf)
- powered. (2018). *powered*. Obtenido de powered: [https://universidad-de-los-andes.gitbooks.io/fundamentos-de-programacion/content/Nivel4/5\\_ManejoDeLasExcepciones.html](https://universidad-de-los-andes.gitbooks.io/fundamentos-de-programacion/content/Nivel4/5_ManejoDeLasExcepciones.html)
- Programaciónestructurada. (2018). *Programaciónestructurada*. Obtenido de Programaciónestructurada: <http://www.utn.edu.ec/reduca/programacion/arreglos/definiciones1.html>
- STREPHONSAYS. (2020). *STREPHONSAYS*. Obtenido de STREPHONSAYS: <https://es.strephonsays.com/difference-between-local-and-global-variable#:~:text=Una%20variable%20global%20es%20una%20variable%20que%20se,el%20valor%20puede%20ser%20cambiado%20por%20otras%20funciones>.