**UNIVERSIDAD DE HUÁNUCO**

“Año del Fortalecimiento de la Soberanía Nacional”

**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

**Programación Orientada a Objetos**

**MONOGRAFÍA**

**Tipos de datos primitivos y estructurados/formatos de salida en java**

**Integrante:**

* NIETO ESPINOZA, FRANK ROLQUIN

**Docente**

ING. Freddy Claydermam Vigilio Arratea

**Huánuco, mayo 2022 - 1**

**INDICE**

[1. Introducción 3](#_Toc102548120)

[2. Tipos de datos primitivos 4](#_Toc102548121)

**2.1 ¿Qué son los tipos de datos primitivos?....................................................................4**

[3. Tipos de datos estructurados 5](#_Toc102548122)

**3.1 Vectores y Matrices …………………………………………………………………………………………….5**

**3.2 Memoria Array …………………………………………………………………………………………………...5**

**3.3 Busqueda Secuencial ……………………………………………………………………………………………5**

**3.4 Cadena de caracteres ……………………………………………………………………………………………5**

**3.5 Estructura …………………………………………………………………………………………………………5**

**3.6 Equivalencia de Tipos ………………………………………………………………………………………5**

4. Formato de Salida en Java…………………………………………………………………………………………6

**4.1 ¿Qué es la clase System ……………………………………………………………………………………………6**

**4.2 Formato de cadena ………………………………………………………………………………………………6**

**4.3 Formato de cadena en fecha y hora …………………………………………………………………………7**

# INTRODUCCIÓN:

El siguiente trabajo fue realizado con el fin de dar a conocer los diferentes tipos de datos, estructuras y formatos, indagando en los diferentes archivos, las cuales pudiéramos obtener la información pertinente, para analizar los tipos de formatos que se puede encontrar en el lenguaje java, formatos que se tiene que escribir si o si con un símbolo de “%”, también se menciona a tipos de datos primitivos, la cual nos ayuda para que todas las variables tengan un tipo de dato asignado y por último se mención a los tipos de datos estructurados que nos ayuda en la búsqueda secuencial, en la cadena de caracteres.

# TIPOS DE DATOS PRIMITIVOS:

* 1. **¿Qué son los tipos de datos primitivos?**

Se define el tipo de dato de la variable a la hora de definir, es por ello que todas las variables tendrán un tipo de dato asignado, el lenguaje javo da de base una serie de tipos de datos primitivos. Como, por ejemplo:

**Byte:** representa un tipo de dato de 8 bits, ósea que puede almacenar valores numéricos. -145 a 145

**Short:** representa un tipo de dato de 16 bits, ósea que puede almacenar valores numéricos, -54.654 a 54.654

**Int:** representa un tipo de dato de 32 bits.

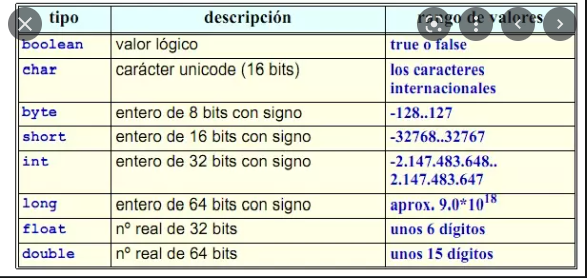
**Long:** representa un tipo de dato de 64 bits.

**Boolean:** sirve para definir tipos de datos booleanos, eso quiere decir aquellos valores que tienen un valor de true o false.

**Char:** representa a un carácter Unicode sencillo de 16 bits.

**Float:** sirve para almacenar números en coma flotante con precisión de 32 bits.

**Double:** sirve para almacenar números en coma flotante con precisión de 64 bits.



# TIPOS DE DATOS ESTRUCTURADOS:

* 1. **VECTORES Y MATRICES:**
     1. **VECTORES:** sirve para agrupar variables de un mismo tipo con un único nombre como, por ejemplo:

Int a0, a1, a2, a3, a4, etc.

otra forma de cómo se puede hacer es utilizando (o array)

la forma exacta de declarar un vector es poniendo:

Tipo Nombre[NumElementos]: es el número de elementos que contiene el valor, siempre y cuando tiene que ser siempre una constante de tipo entero.

* + 1. **MATRICES:** podemos crear vectores de cualquier tipo simple, podemos crear vectores tomando como tipo base de otro vector.
  1. **MEMORIA ARRAY:**

Cuando utilizamos la memoria array, los elementos de un vector se almacenan en memoria de forma contigua, es decir, uno al lado del otro.

Puesto que una matriz no es más que un vector de vectores.

* 1. **BUSQUEDA SECUENCIAL:**

La búsqueda secuencial es el tipo de búsqueda más sencilla y consiste en ir comparando el elemento buscado con todos los elementos del array.

Si el vector que buscamos esta ordenado, la búsqueda puede ser un poco más eficiente, parando en cuanto lleguemos a un elemento que sea mayor que el buscado.

* 1. **CADENA DE CARACTERES:**

Es una secuencia de caracteres que es tratada como un único dato. Por ellos la forma más lógica es con un vector de caracteres.

typedef char cadena[10];

cadena str1, str2;

El entero representa la longitud actual de la cadena, al igual que sucedía con los vectores de tamaño variable que ya hemos visto.

* 1. **ESTRUCTURAS:**

Es una agrupación de elementos de tipos diferentes y en registros. Cada elemento se le denomina campo y se le representa mediante un identificador propio.

La declaración de estructura ya define un tipo.

* 1. **EQUIVALENCIA DE TIPOS:**

Hay algunos tipos de operaciones que sólo se pueden realizar con tipos de datos que sean equivalentes. Estas son la asignación o el pase de parámetros. Por ello es necesario definir el criterio de equivalencia de tipos de datos que sigue un lenguaje determinado.

# **FORMATOS DE SALIDA:**

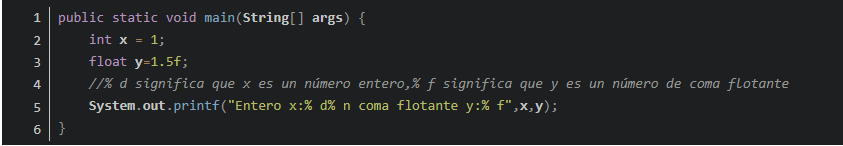
* 1. **¿Qué es la clase System?**
* Es una clase final
* Está en el paquete: java.lang
* Tiene un miembro estatico PrintStream
  1. **FORMATOS DE CADENA:**

# **System.out.printf():** esto sirve para darle un formato a un texto o dato con lo cual estemos trabajando esto se escribe de la siguiente forma:

System.out.printf()

**primero se pone el formato la cual sería así:**

* (“%s”) la cual esto hace referencia a las cadenas o textos.
* (“%d”) la cual esto hace referencia para los números enteros.
* (“%f”) la cual esto hace referencia para los números decimales.
* (“args”) esto hace referencia a los elementos del arreglo.

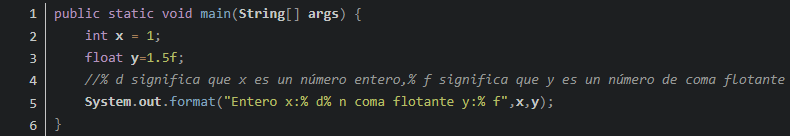


# **System.out.format**(): el método format es para llevar a cabo presentaciones o salidas numéricas e pantalla con el formato apropiado que queramos aplicar, como por ejemplo:

System.out.format

**Primero se pone el formato la cual sería así:**

* (“%%”) este formato hace referencia a que implementa un carácter %
* (“%i”) este formato imprime decimal, uso con enteros.
* (“%u”) este formato hace que imprima un decimal sin signo.
* (“%o”) este formato hace que imprima un numero octal.
* (“%c”) este formato hace que imprima un carácter.
* (“%f”) este formato hace que imprima con una coma flótate.
* (“%a”) este formato hace que el valor hexadecimal del número.
* (“%e”) este formato hace que el número de decimal en notación científica.
* (“%h”) este formato hace que el valor de cadena hexadecimal del método hashCode().



* 1. **FORMATO DE CADENA FECHA HORA TIPOS:**

Cuando usamos co mayúscula junto con % t, obtenemos la salida en mayúscula. Del mismo modo, cuando usamos la minúscula junto con % t, obtenemos la salida en minúscula, por ejemplos:

(“%tC”) este formato hace que el año sea formateado con 2 dígitos.

(“%tc”) fecha y hora en formato.

(“%tD”) fecha en el formato.

(“%td”) día del mes en dos dígitos.

(“%te”) dia del mes sin 0 a la izquierda.

(“%tF”) fecha formateada.