

# Asignatura Introducción a la Programación

## Ingeniería Informática

### Guía de Aprendizaje

**Tema :** Cadenas de caracteres.

**Unidad didáctica:** Cadenas de caracteres. Introducción.

#### 1. Objetivos de la unidad didáctica:

- Identificar las clases que tiene el lenguaje de programación Java para manejar las cadenas de caracteres.
- Caracterizar la clase ***String*** del lenguaje de programación Java.
- Caracterizar la clase ***StringBuffer*** del lenguaje de programación Java.

#### 2. Introducción a la unidad didáctica:

Los caracteres son los bloques de construcción básicos de los programas fuente de Java. Todo programa está compuesto de una secuencia de caracteres que, cuando se agrupan en forma significativa, son interpretados por la computadora como una serie de instrucciones utilizadas para realizar una tarea. Un programa puede contener literales de carácter. Una literal de carácter es un valor entero representado como un carácter entre comillas simples. Por ejemplo, 'z' representa el valor entero de z, y '\n' representa el valor entero de una nueva línea. Es de los tipos de datos que presentan los datos con los cuales se implementa un software en incluso la mayoría de los datos que introduce los usuarios en software son cadenas de caracteres (entiéndase como cadena de caracteres una palabra, nombre, apellidos, un párrafo o texto o una secuencia de caracteres imprimibles desde el teclado). Este tipo de información puede ser almacenada con nuestro conocimientos de programación hasta el momento en un arreglo de tipo de dato **char**.

Esta variante aunque no esta mal hace que el posterior trabajo con este tipo de información o dato sea un tanto compleja. Es por eso que la mayoría de los lenguajes de programación de alto nivel tipado presentan dentro de sus elementos nativos clases que permiten almacenar y manipular este tipo de información de una mejor manera para los programadores. En el caso del lenguaje de programación de Java el mismo cuenta con las clases **String**, **StringBuffer** y **StringBuilder**.

#### 3. Orientaciones para el estudio:

1. Le sugerimos que para realizar el estudio de una materia debe elegir un lugar agradable, limpio, ventilado, cómodo, iluminado y si necesita recursos tecnológicos deben estar disponibles.

2. Cree hábitos de estudio sistemático; para esto:

- Planifique su tiempo y no deje para mañana lo que pueda hacer hoy; una buena planificación hace manejable las responsabilidades diarias que tenemos para con la vida, permite cumplir con todas las tareas programadas y así evita agobios de última hora.
- Calendarice las fechas más importantes de estudio y entrega de tareas. Localice con antelación los materiales que necesita para realizar el estudio individual o colaborativo.

3. Utilice técnicas de estudio:

- Elija un entorno de estudio que resulte agradable y sin elementos que lo que distraigan
- Estudie activamente, para esto lea en voz alta, tome notas, elabore esquemas o mapas conceptuales, realice resúmenes.
- Reflexione sobre lo que va aprendiendo, para esto relacione lo nuevo con lo anterior o conocido, asegúrese de que entiende y es capaz de aplicar lo que está aprendiendo antes de pasar adelante.

4. Interactúe con los materiales de estudio en tres fases:

- Fase de aproximación: Revise el objetivo de la unidad y después la acción o acciones a lograr; busque los materiales sugeridos para desarrollar la tarea, verifique cuál es el tiempo de que dispone.
- Fase de lectura profunda: aproxímese al material a través de una lectura ligera, poniendo especial interés en los títulos y subtítulos. Trate de relacionar lo que va leyendo con conocimientos adquiridos previamente.
- Fase de evaluación: Una vez realizada la lectura, intente realizar las actividades de auto-aprendizaje.

#### 4. Requisitos Previos:

Tener conocimientos básicos de computación: los estudiantes matriculados deben conocer los procedimientos básicos para el manejo de PC y/o dispositivos móviles, aplicaciones de ofimática; así como uso de navegadores de páginas Web y el correo electrónico.

#### 5. Actividades de auto-aprendizajes:

1. Diferencias entre la clase **String**, **StringBuffer** y **StringBuilder** del lenguaje de programación Java.

2. A cuales paquetes de Java pertenecen las clases **String**, **StringBuffer** y **StringBuilder**.

3. Qué ocurre cuando entre dos variables de tipo **String** aplico el operador + ?.

4. Investigue el cual es el máximo de caracteres que soporta cada una de las siguientes clases **String**, **StringBuffer** y **StringBuilder**.

#### 6. Actividades de evaluación:

1. Elabore un cuadro comparativo en cuanto a funcionalidades, semejanzas, diferencias, ventajas y desventajas entre las clases **String**, **StringBuffer** y **StringBuilder**.

#### 7. Resumen:

Las clases **String**, **StringBuffer** y **StringBuilder** presente en el lenguaje de programación en Java muchas veces se confunden con tipos de datos nativos del lenguaje cuando no es así. Lo cabe duda es la utilidad que tienen dichas clases para el trabajo y almacenamiento de las cadenas de caracteres hacen que muchos de los programadores la tomen como datos nativos del lenguaje.

#### 8. Glosario de términos:

Consulte la bibliografía y defina los siguientes conceptos:

- Clase (en programación)

#### 9. Bibliografía:

Aprenda Java como si estuviera en primero. *Colectivo de autores*. Cap 4 epígrafes 4.2.

Cómo programar en Java. *Deitel, Paul J. Y Harvey M. Deitel*. Cap 30 epígrafes 30.1, 30.2, 30.3 y 30.4.

#### 10. Recursos educativos digitales RED:

#### 11. Próxima unidad: Clase **String**.