



Recibe una cálida:

# ¡Bienvenida!

---

Te estábamos esperando 😊 

# ➤ El Entorno Java para la programación

---

**Plan formativo:** Desarrollo de Aplicaciones Full Stack Java Trainee V2.0

# HOJA DE RUTA

¿Cuáles **skill** conforman el programa?



# REPASO CLASE ANTERIOR

En la clase anterior trabajamos :

- ✓ *Comprender la sintaxis e implementación de los arreglos*
- ✓ *Reconocer la importancia de las colecciones*

# LEARNING PATHWAY

4.3

Start! 🏁

El Entorno Java para  
la programación

Clase String

Sin Cadenas

Clases de utilidad: String

# OBJETIVOS DE APRENDIZAJE

¿Qué aprenderemos?



***Conocer y manipular las funciones asociadas a la clase String***





# Rompehielo 🧊

## Cadena literaria: 🙌



A través del chat, el moderador irá pidiendo a los participantes que propongan una palabra corta que rime con la anterior, por ejemplo: Me gusta la programación. Levantando la mano, deben nombrar palabras que rimen con programación. El moderador irá escribiendo en pantalla las palabras que nombremos. Luego el moderador pegará la cadena completa en el chat y resolveremos algunas preguntas:



## Respondan en el chat: 🖋️

- 1- ¿Cómo podríamos insertar una nueva palabra al inicio?
- 2- ¿Cómo podríamos contar la cantidad de vocales que hay en la frase?
- 3- ¿Los espacios en blanco, son caracteres?

# › Clase de utilidad: String





# Clase String



## ¿Qué es la Clase String?:

La clase String en Java es una herramienta poderosa **para manipular cadenas de texto**. Con ella puedes concatenar cadenas para unir las, obtener la longitud de una cadena, buscar subcadenas, reemplazar partes de una cadena y mucho más. Además, la clase String **proporciona métodos** para comparar cadenas y realizar operaciones de formato.



La Clase String tiene muchos métodos para manipular cadenas, en ésta clase veremos algunos de ellos.





# String: concatenar

## Concatenación de Strings:



La concatenación de Strings en Java se puede realizar con el método **concat()** que pertenece a la propia clase String.

Con **'texto1.concat(" ")'** le indicamos que queremos concatenar el espacio " " al valor del string de **'texto1'**. Luego, **con un punto, podemos ir añadiendo más concat()**. En este caso, se **añade el valor de 'texto2'**.



```
7      String texto1 = "Hola";
8      String texto2 = "Mundo";
9      String texto3 = texto1.concat(" ").concat(texto2);
10
11      System.out.print(texto3);|
```

<

Console X

<terminated> main [Java Application] C:\Users\marie\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.  
Hola Mundo



# String: comparar

## Comparación de Strings:



La comparación de Strings en Java se puede realizar con el método **equals()** que pertenece a la propia clase String.

Con **'texto1.equals(texto2)'** le indicamos que queremos comparar si ambas cadenas son iguales. Esta función nos devuelve un valor booleano.



```
7      String texto1 = "Hola";
8      String texto2 = "Mundo";
9      Boolean iguales = texto1.equals(texto2);
10
11      System.out.print(iguales);
12  }
```

Console X

<terminated> main [Java Application] C:\Users\marie\.p2\pool\plugins\org.eclipse.justj.openj  
false



# String: índices

## Concatenación de Strings:

Los Strings están formados por un conjunto variable de caracteres. Entonces, para poder manipular individualmente cada uno de estos, existe lo que se conoce como índice. El índice de los Strings **empieza por el valor entero 0** y va aumentando en 1 con cada caracter.

**El método indexOf()** devuelve la posición de la primera ocurrencia de una subcadena en una cadena.



```
4
5 public static void main(String[] args) {
6
7     String texto1 = "Hola Mundo";
8     int valorIndice = texto1.indexOf("Mundo");
9
10    System.out.print(valorIndice);
11 }
```

Console X

<terminated> main [Java Application] C:\Users\marie\.p2\pool\plugins\org.eclipse.justj.openjdk

5



# String: reemplazar

## El método replace()

El método replace() reemplaza todas las ocurrencias de una subcadena con otra subcadena. Se esperan dos argumentos en la llamada. Sencillamente, se busca el primer valor y se reemplaza en todas las ocurrencias.



```
6
7   String texto1 = "hola, este String tiene más de un hola. Está hecho para probar "
8       + "el metodo replace con la palabra hola.";
9
10  String nuevoTexto = texto1.replace("hola", "adiós");
11
12  System.out.print(nuevoTexto);
13 }
```

Console X

<terminated> main [Java Application] C:\Users\marie\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86\_64\_17.0.7.v20230425-1502\jre\bin\javaw.exe (16 ago. 2023)

adiós, este String tiene más de un adiós. Está hecho para probar el metodo replace con la palabra adiós.



# String: subcadena

## El método substring()

El método substring() devuelve una subcadena que **empieza desde la posición indicada** como argumento.



```
7      String texto1 = "Hola Mundo";
8      String character = texto1.substring(2);
9
10     System.out.print(character);
11 }
12
13 }
```

Console X

<terminated> main [Java Application] C:\Users\marie\.p2\pool\plugins\org.eclipse.justj.openjdk.hotsp

la Mundo



# LIVE CODING

Ejemplo en vivo

## Manipulando Strings:

*Vamos a ver el funcionamiento de algunas funciones de Strings que pueden ser muy útiles a la hora de desarrollar nuestros programas.*

- 1. Realizar un programa que solo permita introducir frases o palabras de **8 de longitud**. Si el usuario ingresa una cadena de 8 de longitud se deberá imprimir un mensaje por pantalla que diga “CORRECTO” e **imprimir la frase en mayúsculas**, en caso contrario, se deberá imprimir “INCORRECTO” y toda **la frase en minúsculas**.*

**Tiempo: 25 minutos**



# **Ejercicio N° 1**

# **Sin cadenas**





# Sin cadenas

## Contexto: 🙌

La manipulación de cadenas es algo muy habitual en la programación. Se utiliza tanto para comparar contraseñas, validar usuarios, y controlar los datos que se ingresan por teclado. Es por eso que pondremos en práctica algunos de los conceptos aprendidos.

## Consigna: 🛠️

- 1- Dada una cadena, extraer la cuarta y quinta letra usando el método substring.
- 2- Dada una cadena mostrar por pantalla la cantidad de vocales que tiene. Ejemplo:

**Entrada:** cad = "Hola tu"

**Salida:** La cantidad de vocales es 3

**Tiempo** 🕒: 30 minutos

## Tips: ⚙️

- 1- Recuerda que los índices comienzan en 0.
- 2- Puedes recorrer la cadena utilizando un ciclo for, y contar las vocales con condicionales y una variable acumuladora (int)

○

# ¿Alguna consulta?

+



# RESUMEN

¿Qué logramos en esta clase?

- ✓ **Comprender la importancia e implementación de la clase de utilidad `String` y sus métodos asociados.**



# #WorkingTime

Continuemos ejercitando

¡Antes de cerrar la clase! Te invitamos a: 📌 📌 📌

1. Repasar nuevamente la grabación de esta clase
2. Revisar el material compartido en la plataforma de Moodle (lo que se vio en clase y algún ejercicio adicional)
  - a. Material 1 (Foro)
  - b. *Lectura Módulo 4, Lección 3: páginas 13 - 15*
3. Traer al próximo encuentro, todas tus dudas y consultas para verlas antes de iniciar nuevo tema.

# ¡Muchas Gracias!

Nos vemos en la próxima clase 🙌



Momento: ✚

# Time-out!

🕒 5 min.

