

# TALLER DE LENGUAJE DE PROGRAMACIÓN I

## Entorno de CONSOLA y GRÁFICOS

### UNIDAD 1

#### Estructuras de datos primitivos de JAVA

JAVA

# OBJETIVO

- Creación de archivos JAVA
- Datos primitivos en JAVA
  - Enteros  
Int, Short, Long, Byte
  - Coma Flotantes (Decimales)  
Float, Double
  - Carácter
  - Lógicos (Booleanos)

## Trabajo Independiente

- Identificar IDE en los que puedes desarrollar en JAVA - Eclipse.
- Crear una aplicación, donde se utilicen todos los tipos de datos y se realicen las operaciones aritméticas +, -, \*, /

# Creación de archivos JAVA

- Lo primero es contar con:
  - JDK Instalado (verificarlo)
    - Ejecutar cmd
    - Escribir java o javac
  - Editor de archivos de texto (Script)

# Creación de archivos JAVA

- Creemos nuestro primer programa JAVA

Bienvenidos TLPI 2023-01!

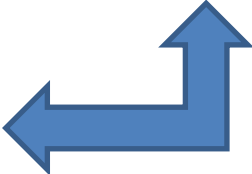
```
public class MiPrimeraAplicacion
{
    public static void main (String[] args)
    {
        System.out.println("Hola grupo!");
    }
}
```

# Creación de archivos JAVA, continuación

## Exigencia de JAVA

- Garantizar que el nombre del archivo tenga el mismo nombre de nuestra “**clase**”

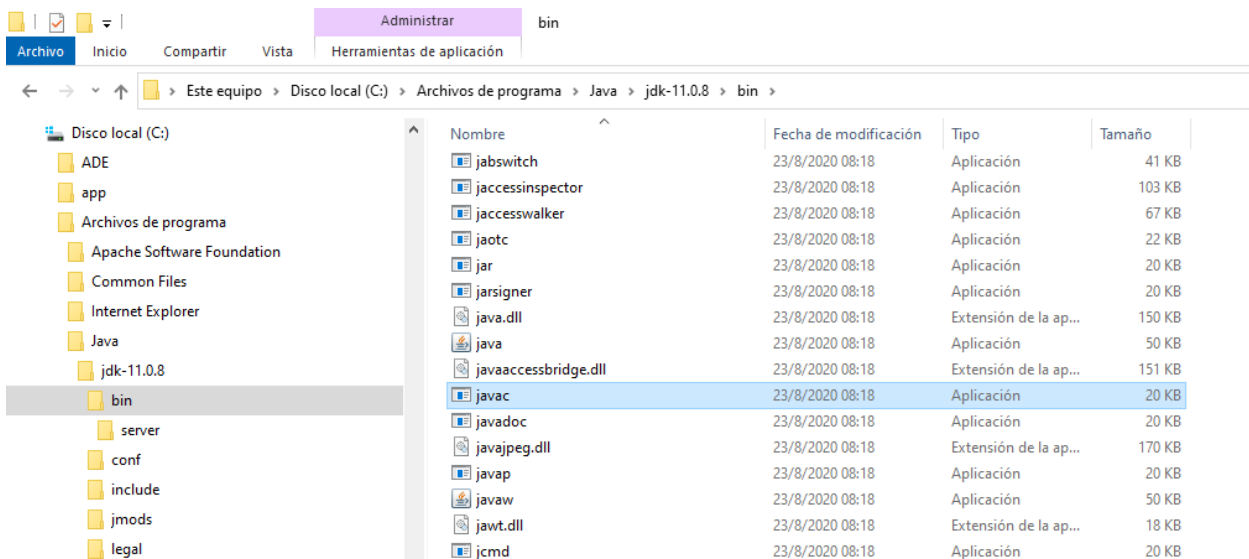
```
public class MiPrimeraaplicacion
{
    public static void main (String[] args)
    {
        System.out.println("Hola grupo!");
    }
}
```



# Creación de archivos JAVA, continuación

## Compilador de JAVA

- El nombre del compilador es javac

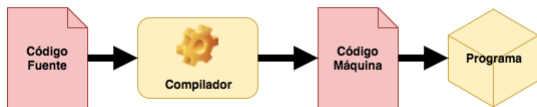


# Creación de archivos JAVA, continuación

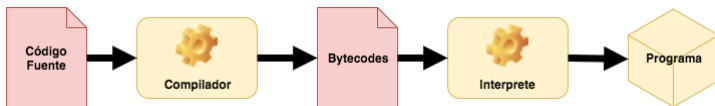
## Compilador de JAVA

- Compilemos nuestro primer programa
  - Abrimos la ventana cmd
  - Ubicarse en el directorio donde se guardó el archivo .java
  - Ejecutamos el comando **d:/javac MiPrimeraAplicacion.java**

### Compilación C



### Compilación JAVA



```
Administrador: Símbolo del sistema

D:\>javac MiPrimeraAplicacion.java

D:\>dir
El volumen de la unidad D no tiene etiqueta.
El número de serie del volumen es: 3453-4DE7

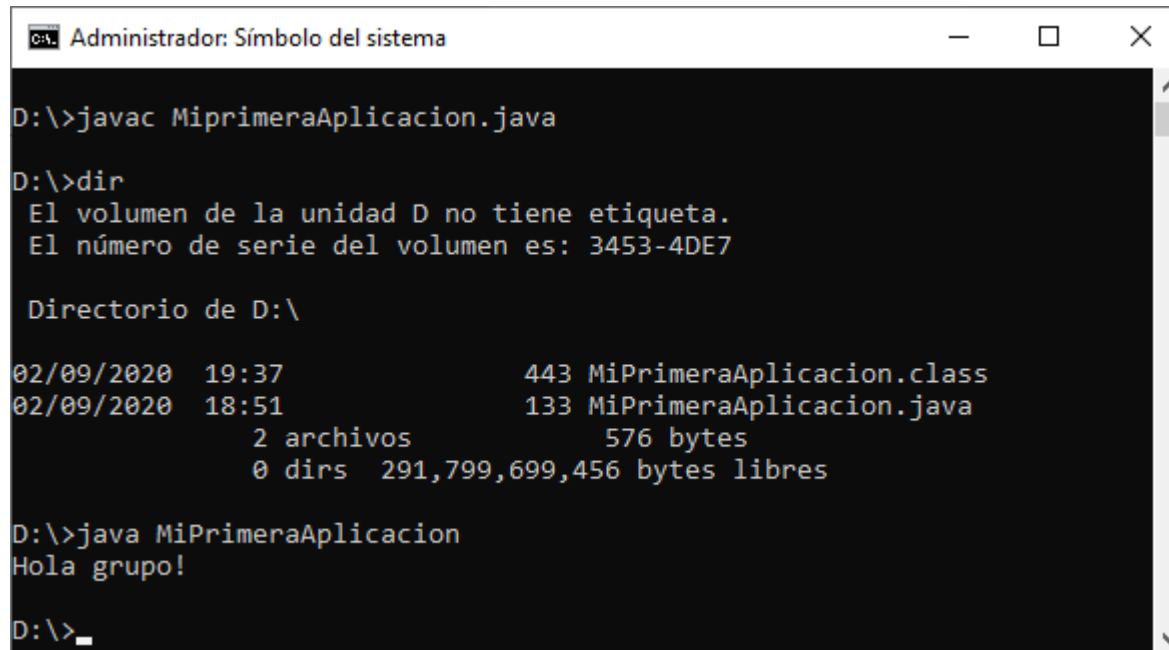
Directorio de D:\

02/09/2020  19:37                443 MiPrimeraAplicacion.class
02/09/2020  18:51                133 MiPrimeraAplicacion.java
                2 archivos                576 bytes
                0 dirs  291,799,699,456 bytes libres

D:\>
```

# Creación de archivos JAVA, continuación

- Ejecutar nuestro primer programa
  - Ejecutar el comando `d:/java MiPrimeraAplicacion`



```
Administrador: Símbolo del sistema

D:\>javac MiPrimeraAplicacion.java

D:\>dir
El volumen de la unidad D no tiene etiqueta.
El número de serie del volumen es: 3453-4DE7

Directorio de D:\

02/09/2020  19:37                443 MiPrimeraAplicacion.class
02/09/2020  18:51                133 MiPrimeraAplicacion.java
                2 archivos                576 bytes
                0 dirs  291,799,699,456 bytes libres

D:\>java MiPrimeraAplicacion
Hola grupo!

D:\>_
```



# Datos primitivos en JAVA

## Variables en Java

Las variables Java son un espacio de memoria en el que guardamos un determinado valor (o dato).

Para definir una variable seguiremos la estructura:

```
[privacidad] tipo_dato identificador;
```

**Las variables en JAVA deben ser inicializadas,  
para poder ser utilizadas.**

# Datos primitivos en JAVA, continuación

## Tipos de datos primitivos en Java

### Enteros

- Int
- Short
- long
- Byte

### Coma flotante (Decimales)

- Float
- Double

Char

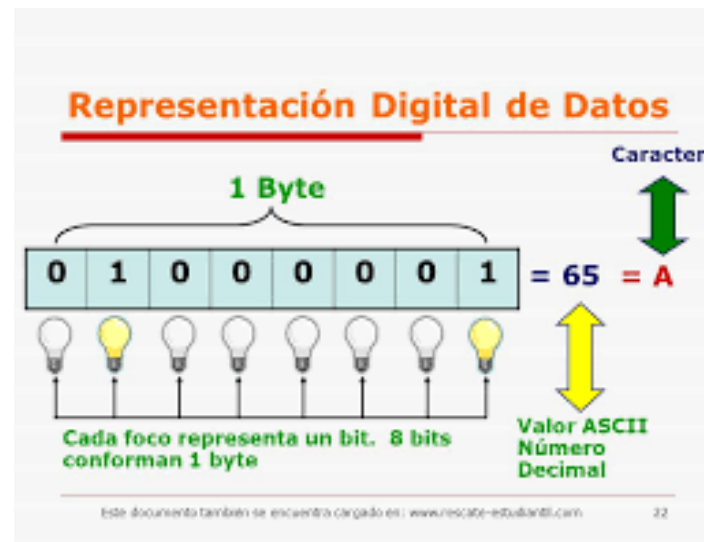
boolean

# Datos primitivos en JAVA, continuación

## Tipos de datos primitivos en Java

### Definición de byte, bit y su representación

Byte es la unidad de información de base utilizada en computación y en telecomunicaciones, y que resulta equivalente a un conjunto ordenado de ocho bits, equivalente al termino de octeto (ocho)

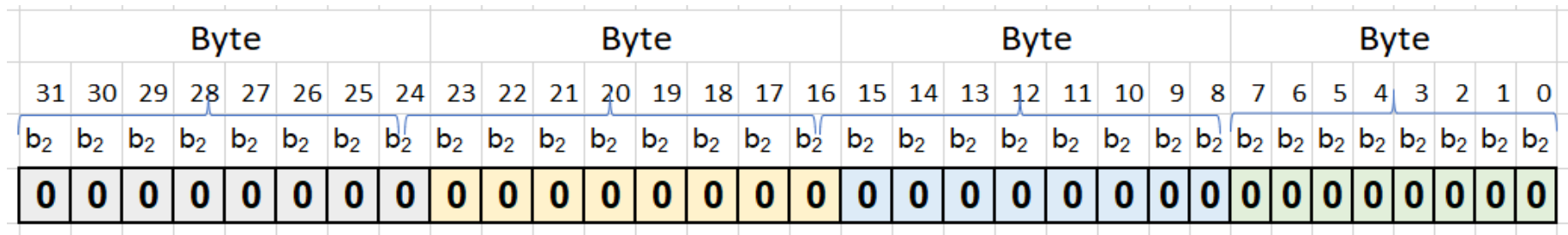


# Datos primitivos en JAVA, continuación

## Tipos de datos primitivos en Java

### int

Se utilizan 4 byte, representados en 32 bits con signo para almacenar valores numéricos. Cuyo valor mínimo es -2,147,483,648 hasta 2,147,483,647



Hacer la representación del paso de tener los 4 byte con todos los bit prendidos en base 10

# Datos primitivos en JAVA, continuación

## Ejemplo definición de variable tipo int

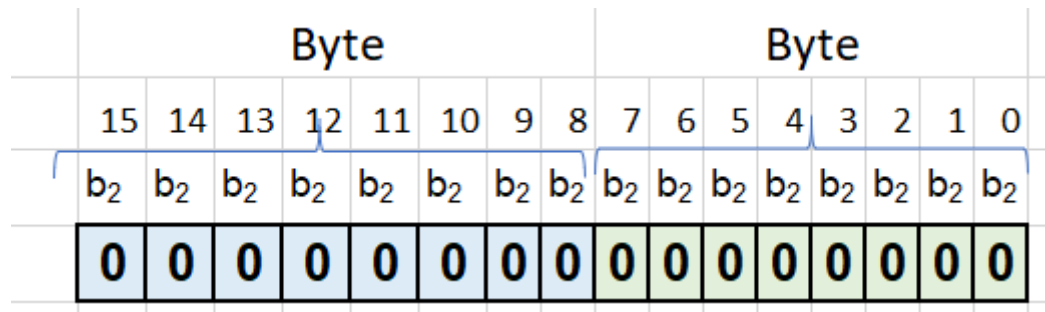
```
public class DatosPrimitivos
{
    public static void main (String[] args)
    {
        int dato = 0;
        System.out.println("Valor de dato:" + dato);
        dato = 15000;
        System.out.println("Valor de dato:" + dato);
    }
}
```

# Datos primitivos en JAVA, continuación

## Tipos de datos primitivos en Java

### Short

Se utilizan 2 byte, representados en 16 bits con signo para almacenar valores numéricos. Cuyo valor mínimo es -32,768 hasta 32,767



Hacer la representación del paso de tener los 2 byte con todos los bit prendidos en base 10

# Datos primitivos en JAVA, continuación

## Ejemplo definición de variable tipo short

```
public class DatosPrimitivos
{
    public static void main (String[] args)
    {
        short dato = 0;
        System.out.println("Valor de dato:" + dato);
        dato = 3300;
        System.out.println("Valor de dato:" + dato);
    }
}
```

# Datos primitivos en JAVA, continuación

## Tipos de datos primitivos en Java

### Long

Se utilizan 8 byte, representados en 64 bits con signo para almacenar valores numéricos. Para su definición se utiliza el sufijo L

-9223372036854775808 a 9223372036854775807



# Datos primitivos en JAVA, continuación

## Ejemplo definición de variable tipo long

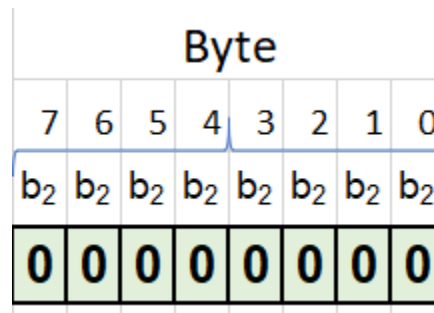
```
public class DatosPrimitivos
{
    public static void main (String[] args)
    {
        long dato = 0;
        System.out.println("Valor de dato:" + dato);
        dato = 5000000000000L;
        System.out.println("Valor de dato:" + dato);
    }
}
```

# Datos primitivos en JAVA, continuación

## Tipos de datos primitivos en Java

### Byte

Se utilizan 1 byte, representados en 8 bits con signo para almacenar valores numéricos. Cuyo valor mínimo es -128 hasta 127



# Datos primitivos en JAVA, continuación

## Ejemplo definición de variable tipo byte

```
public class DatosPrimitivos
{
    public static void main (String[] args)
    {
        byte dato = 0;
        System.out.println("Valor de dato:" + dato);
        dato = 127;
        System.out.println("Valor de dato:" + dato);
    }
}
```

# Datos primitivos en JAVA, continuación

## Tipos de datos primitivos en Java

### Float

Se utilizan 4 byte, representados en 32 bits con signo para almacenar valores numéricos, con capacidad de manejar entre 6 y 7 decimales. Para su definición se utiliza el sufijo F

# Datos primitivos en JAVA, continuación

## Ejemplo definición de variable tipo float

```
public class DatosPrimitivos
{
    public static void main (String[] args)
    {
        float dato = 0.0f;
        System.out.println("Valor de dato:" + dato);
        dato = 10.5f;
        System.out.println("Valor de dato:" + dato);
    }
}
```

# Datos primitivos en JAVA, continuación

## Tipos de datos primitivos en Java

### Double

Se utilizan 8 byte, representados en 32 bits con signo para almacenar valores numéricos, con capacidad de manejar aproximadamente 15.

# Datos primitivos en JAVA, continuación

## Ejemplo definición de variable tipo double

```
public class DatosPrimitivos
{
    public static void main (String[] args)
    {
        double dato = 0.0f;
        System.out.println("Valor de dato:" + dato);
        dato = 10.123456789;
        System.out.println("Valor de dato:" + dato);
    }
}
```

# Datos primitivos en JAVA, continuación

## Tipos de datos primitivos en Java

Char

Se utilizan 1 byte, representados en 8 bits.



# Datos primitivos en JAVA, continuación

## Ejemplo definición de variable tipo char

```
public class DatosPrimitivos
{
    public static void main (String[] args)
    {
        char dato = 'a';
        System.out.println("Valor de dato:" + dato);
        dato = 'b';
        System.out.println("Valor de dato:" + dato);
    }
}
```

# Datos primitivos en JAVA, continuación

## Tipos de datos primitivos en Java

### Boolean

Se utiliza 1 byte, representados en 8 bits, define tipos de datos verdaderos o falsos.

# Datos primitivos en JAVA, continuación

## Ejemplo definición de variable tipo boolean

```
public class DatosPrimitivos
{
    public static void main (String[] args)
    {
        boolean dato = true;
        System.out.println("Valor de dato:" + dato);
        dato = false;
        System.out.println("Valor de dato:" + dato);
    }
}
```

# Trabajo Independiente

- Identificar IDE en los que puedes desarrollar en JAVA.
- Crear una aplicacion, donde se utilicen todos los tipos de datos númericos donde se realicen las operaciones aritméticas +, -, \*, /