

# Proyectos IV

Diseño Industrial

# Unidad 1: Interfaces digitales

## *Objetivos de la unidad*

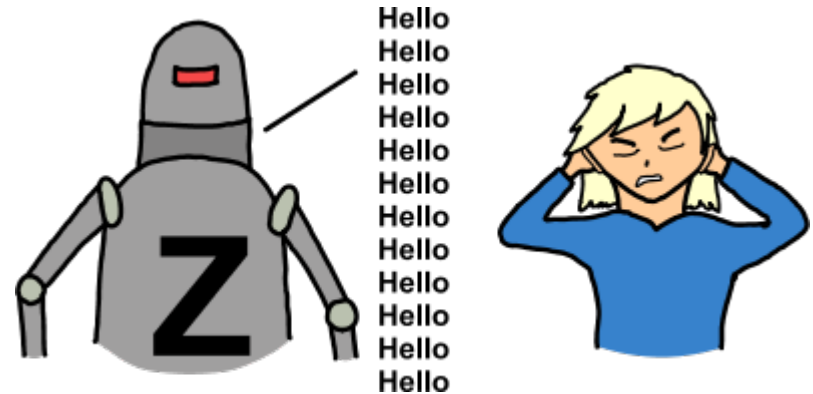
- Comprender los requerimientos básicos para el análisis de problemas que involucren el pensamiento secuencial o algorítmico.
- Entender los fundamentos de programación para la creación de interfaces digitales navegables sensibles al usuario.

# Estructuras de repetición

- Semana 3 -

# ¿Qué son las estructuras de repetición?

Sometimes you want a computer to repeat something over and over again. A loop is used to make a computer do something more than one time.



# ¿En qué casos es útil una repetitiva?

Existe un patrón a ejecutar varias veces con pequeñas variaciones

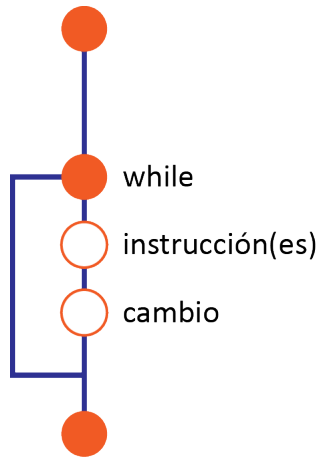
Se requiere ejecutar la misma tarea  $n$  número de veces

# Concepto

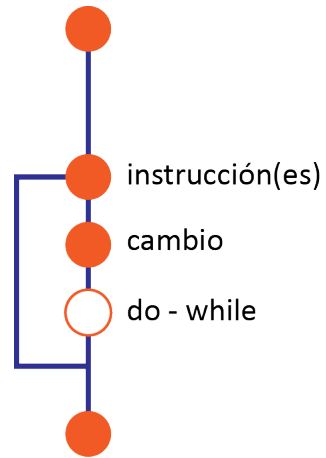
**1**  
**2**  
**3**  
**4**  
**5**  
**6**  
**7**  
**8**  
**9**  
**10**

Las estructuras repetitivas,  
buscan simplificar  
(nuevamente simplificar el  
manejo de patrones que  
siguen un patrón que se  
puede predecir (repeticiones,  
incrementos, distribuciones).

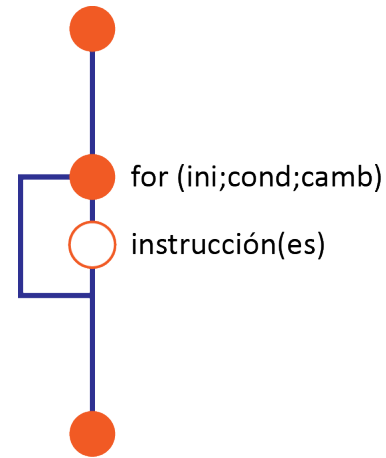
# Repetición en Java



```
while(cond){  
  // instrucciones  
}
```



```
do{  
  // instrucciones  
}while(cond)
```



```
for(ini ; cond ;  
camb){  
  // instrucciones  
}
```

# La condición (booleana)

1. La forma básica de todas las estructuras de repetición requiere de una condición que determina si el ciclo se ejecuta o no.
2. Existen algunas variables asociadas a la palabra break

Recuerde los operadores de comparación

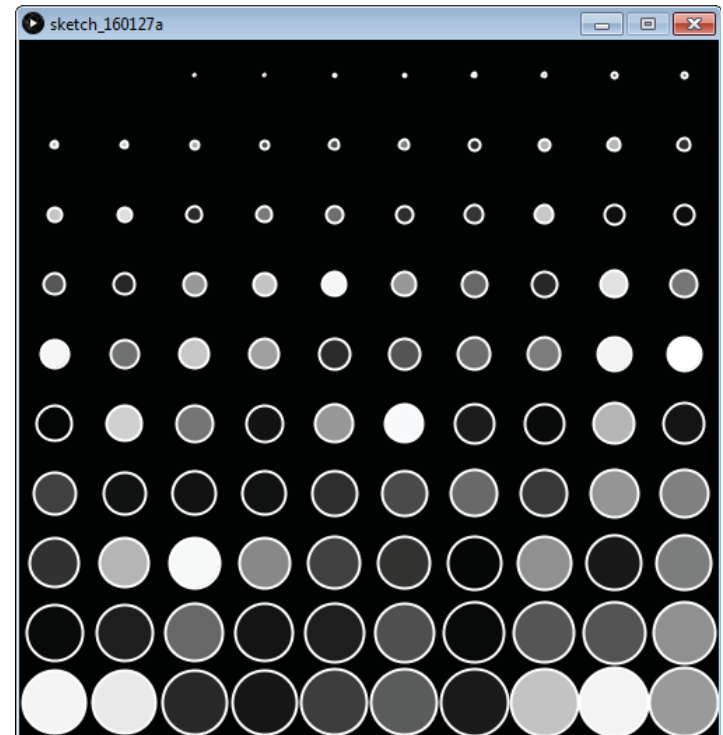
== != > < >= <=



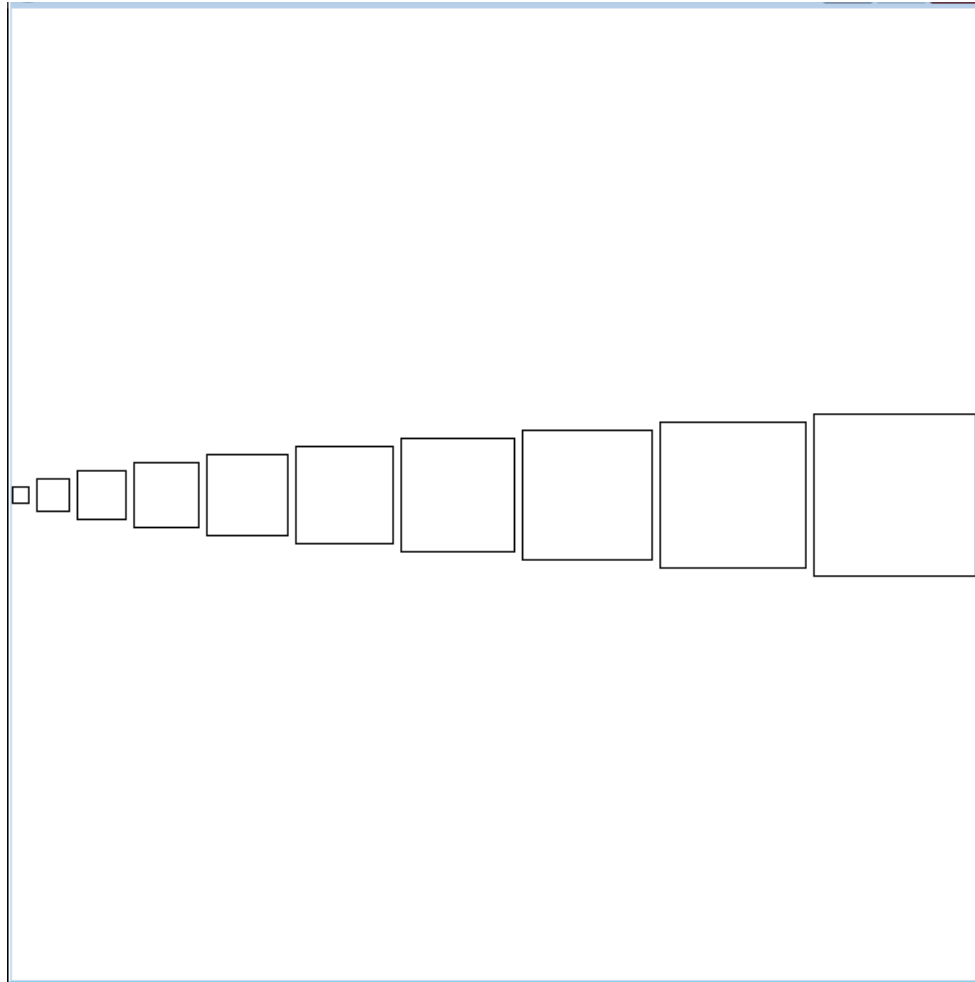
# Ejemplo con for

es la más usada de las estructuras repetitivas, su ciclo depende del parámetro condición que se le debe entregar al igual que el while, pero incluye dentro de la declaración todos los componentes necesarios para la ejecución del mismo (inicialización, condición, incremento).

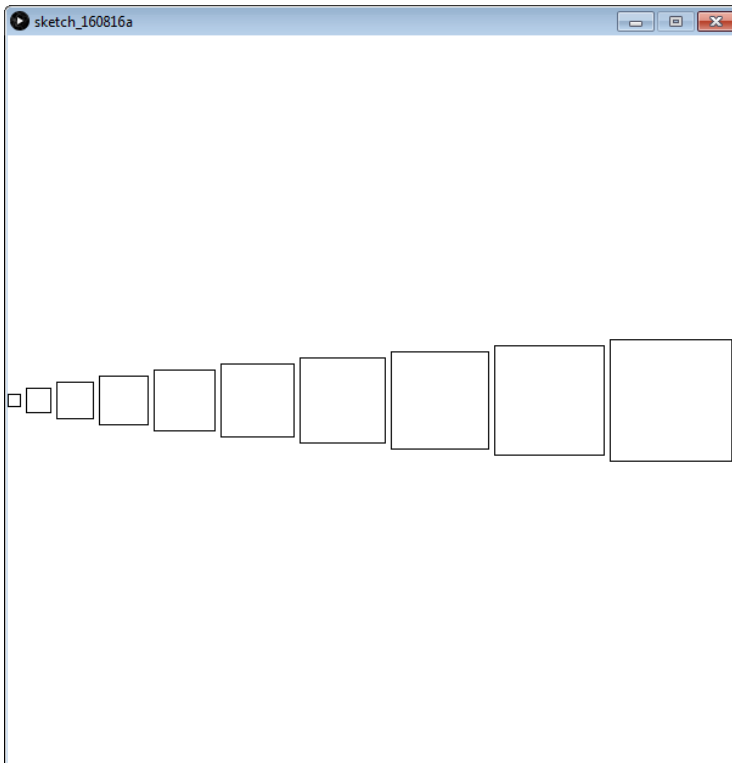
```
int x = 25;
int y = 25;
for(int i = 0 ; i < 100; i++){
    fill(random(255));
    ellipse(x,y,i/2,i/2);
    x+=50;
    if(x>=width-10){
        x=25;
        y+=50;
    }
}
```



# ¿Cómo logramos esto con for?



# Solución



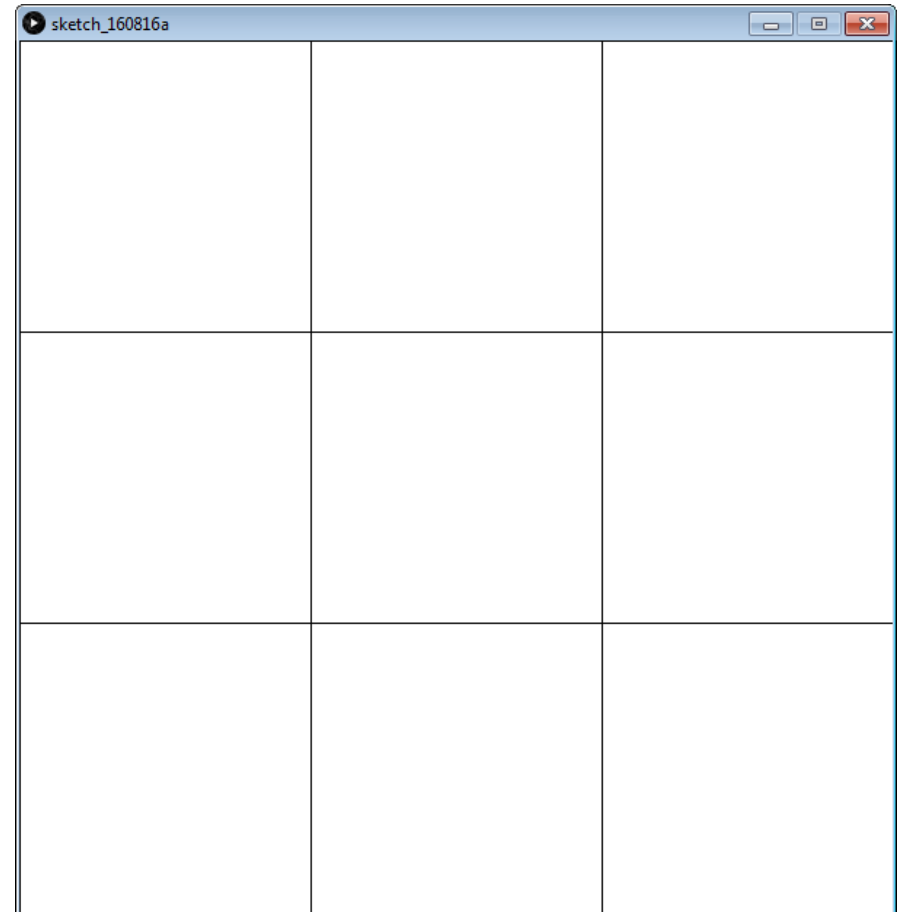
```
void setup(){  
  size(600,600);  
  rectMode(CENTER);  
}  
  
void draw(){  
  background(255);  
  int x = 5;  
  int y = height/2;  
  int tam = 10;  
  for(int i = 0 ; i < 10 ; i++){  
    rect(x, y, tam, tam);  
    tam+=(100/10);  
    x+=tam;  
  }  
}
```

# Ejercicio en Clase

Semana 3

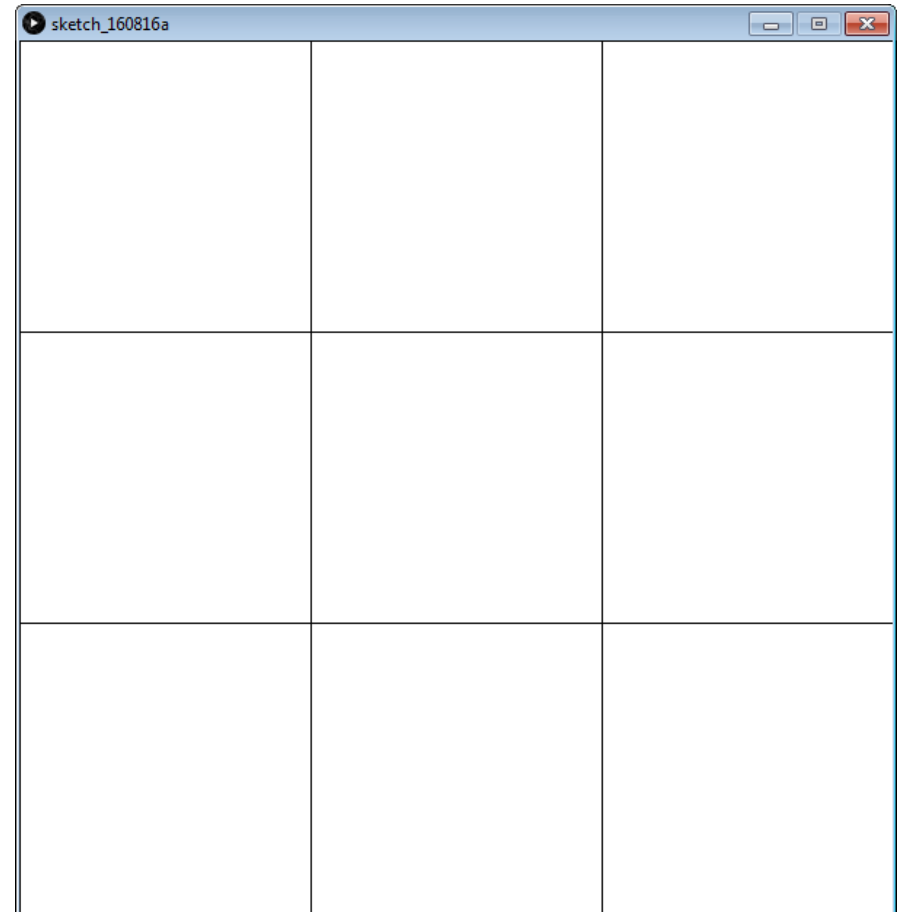
# Ejercicio en Clase (parte 1)

- Debe crear en una nueva aplicación, (Lienzo de 600x600) con 9 rectángulos de tamaño 200x200 que forman una cuadrícula de 3x3.



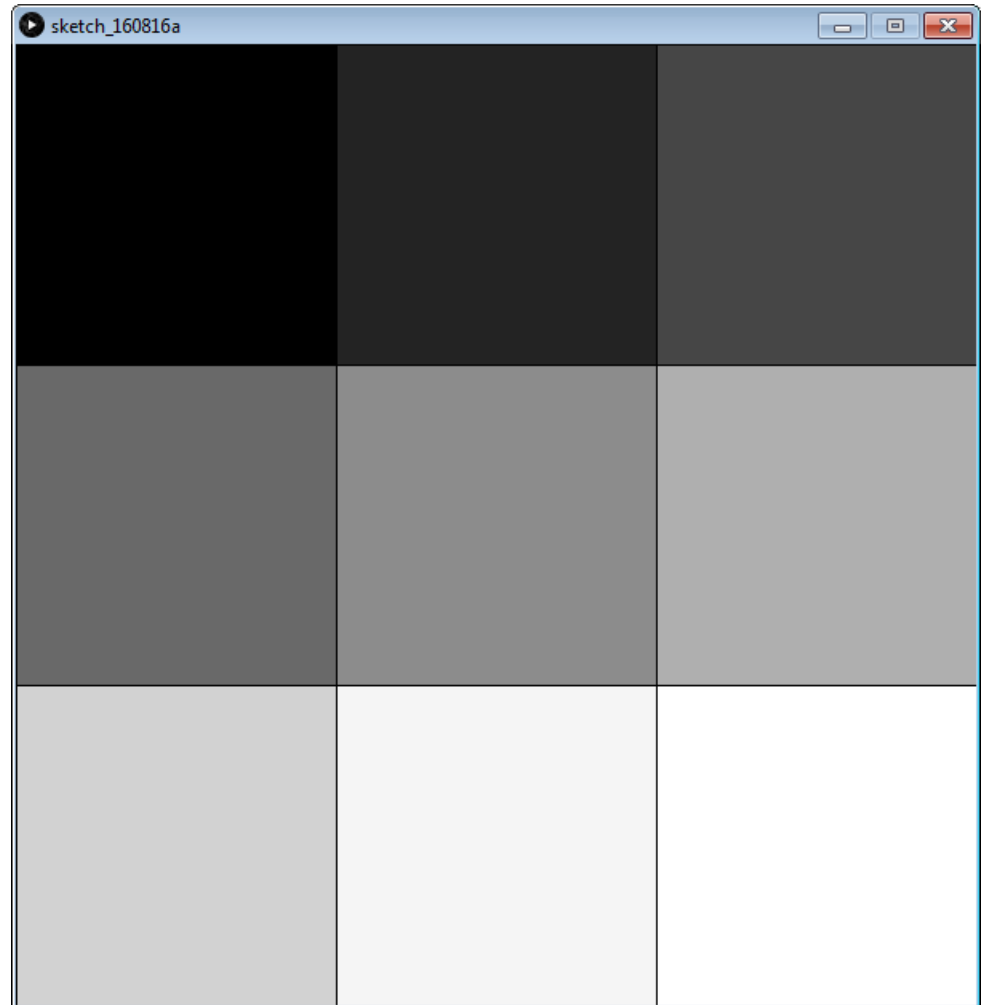
# Ejercicio en Clase (parte 1 - solución)

```
void setup(){  
  size(600,600);  
}  
void draw(){  
  background(255);  
  int x = 0;  
  int y = 0;  
  int tam = 200;  
  for(int i = 0 ; i < 9 ; i++){  
    rect(x, y, tam, tam);  
    x+=tam;  
    if(x>=width){  
      x = 0;  
      y+=tam;  
    }  
  }  
}
```



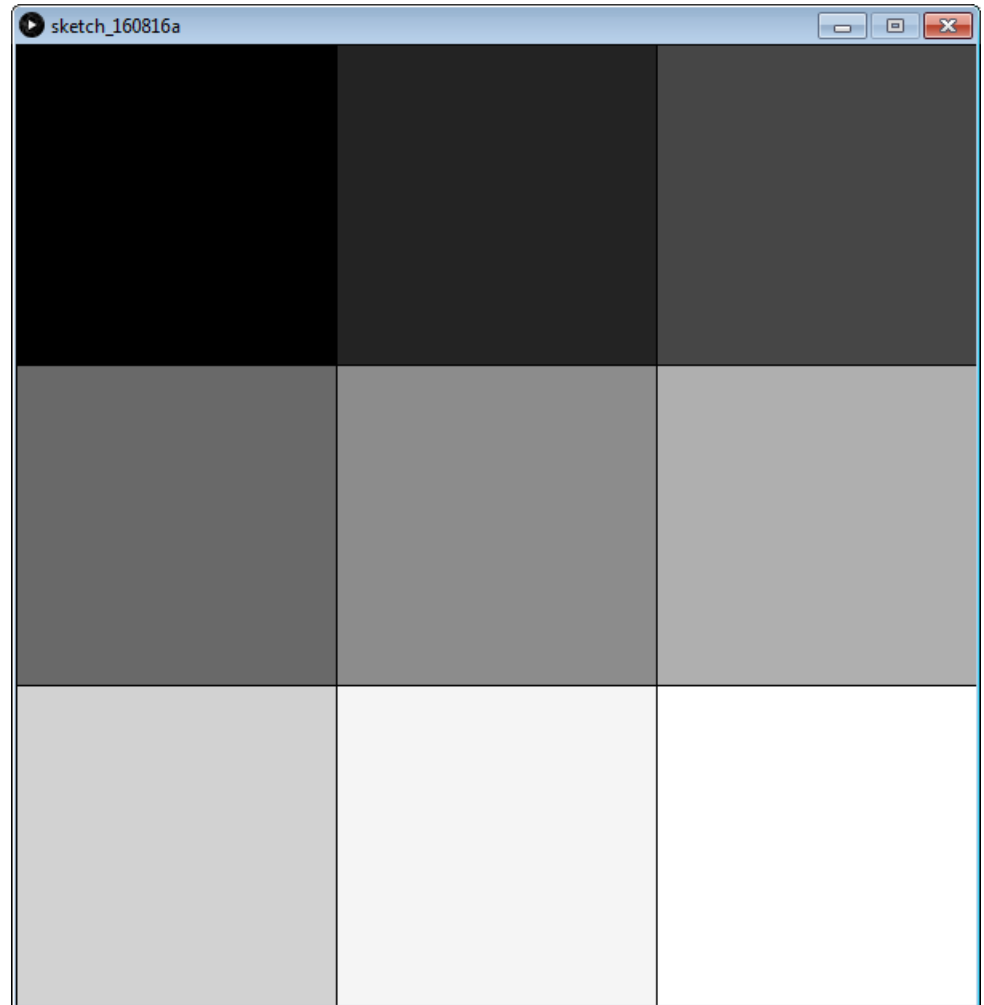
# Ejercicio en Clase (parte 1)

- Debe crear en una nueva aplicación, (Lienzo de 600x600) con 9 rectángulos de tamaño 200x200 que forman una cuadrícula de 3x3.
- Incluya las estructuras que se requieran para generar la imagen de referencia.



# Ejercicio en Clase (parte 1 - Solución)

```
void setup(){
  size(600,600);
}
void draw(){
  background(255);
  int x = 0;
  int y = 0;
  int tam = 200;
  int col = 0;
  for(int i = 0 ; i < 9 ; i++){
    fill(col);
    rect(x, y, tam, tam);
    x+=tam;
    if(x>=width){
      x = 0;
      y+=tam;
    }
    col += 35;
  }
}
```





# Arreglos

- Semana 3 -

# ¿Qué son los arreglos?

- Agrupaciones que almacenan datos del mismo tipo.
- Pueden ser de datos primitivos (int, float, char)
- Pueden ser de datos referenciados (String)

# Ejemplo ...

```
int[] valores;  
valores = new int[5];  
valores[0] = 1;  
valores[1] = 2;  
valores[2] = 3;  
valores[3] = 4;  
valores[4] = 5;
```

```
for(int i = 0; i < valores.length; i++){  
    valores[i] = i+1;  
}
```

```
for(int i = 0; i < valores.length; i++){  
    println("Valor en "+i+" : "+ valores[i]);  
}
```

Los arreglos son agrupaciones variables, la utilización de los mismos facilita(simplifica) el manejo de un mayor numero de datos de forma más sencilla al mismo tiempo que disminuye el número de líneas de código.

```
Valor en 0 : 1  
Valor en 1 : 2  
Valor en 2 : 3  
Valor en 3 : 4  
Valor en 4 : 5
```

# Métodos

- Semana 3 -

# ¿Qué son los métodos?

- Procedimientos que deseo repetir n número de veces
- Cálculos que dependen de entradas fijas
- Validaciones que requieren una respuesta (int, boolean, float, char, String ... etc)

# Equipos

**Equipos de 3 personas**

**Enlace en Moodle.**

# Selección de Usuarios (Proyecto 1)

1. Esquimal
2. Vikingo
3. Samurái
4. Muisca
5. Ciudadano del imperio Romano
6. Ciudadano de una colonia lunar
7. Azteca
8. Ninja
9. Señor feudal
10. Cruzado
11. Sarraceno

# Tarea

- Capitulo 4 - Programación Orientada a Objetos
- Propuesta para Proyecto I – Interfaz de Control de Vivienda por usuario.

[https://www.youtube.com/user/shiffman/playlists?shelf\\_id=2&view=50&sort=dd](https://www.youtube.com/user/shiffman/playlists?shelf_id=2&view=50&sort=dd)