# Sesión 03: Clases

## Programación 2

Ángel Herranz

Febrero 2019

Universidad Politécnica de Madrid

# En capítulos anteriores

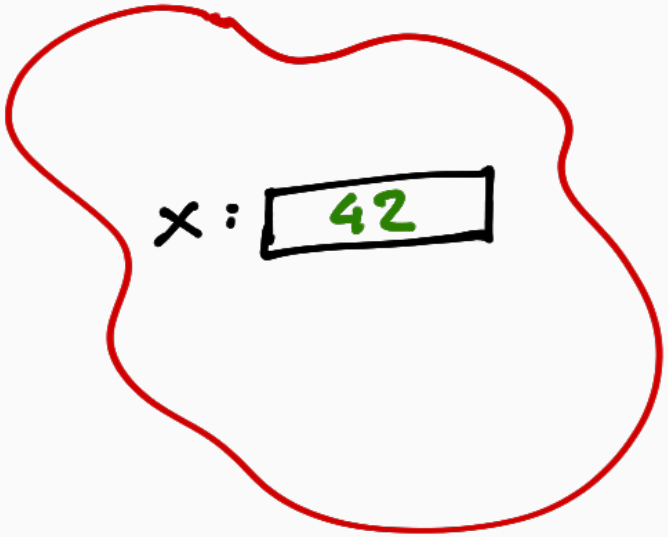
- Sobre los IDEs
- Clases y objetos (intro)
- Objetos, referencias y variables (y primitivos)

# Objetos, referencias y variables

```
public class Sesion02 {
  public static void main(String args[]) {
    A a;
    a = new A();
  }
}
```

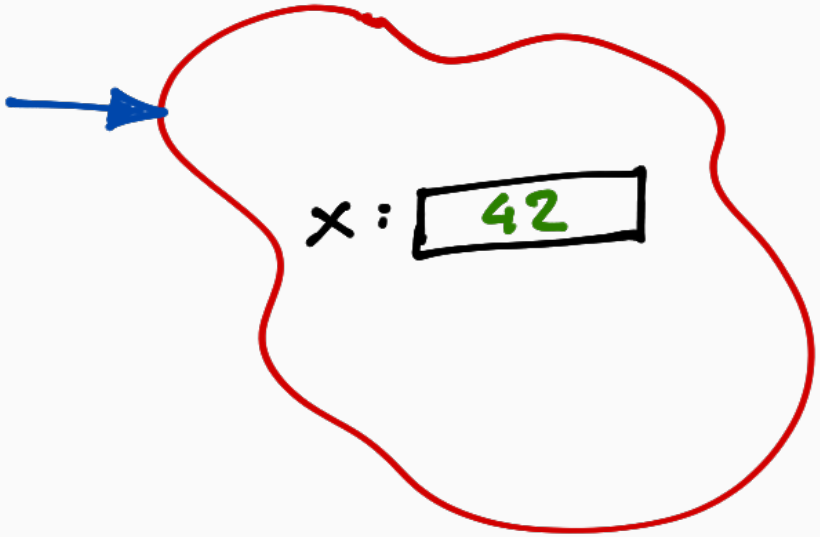
# Objetos, referencias y variables

```
public class Sesion02 {
  public static void main(String args[]) {
    A a;
    a = new A();
  }
}
```

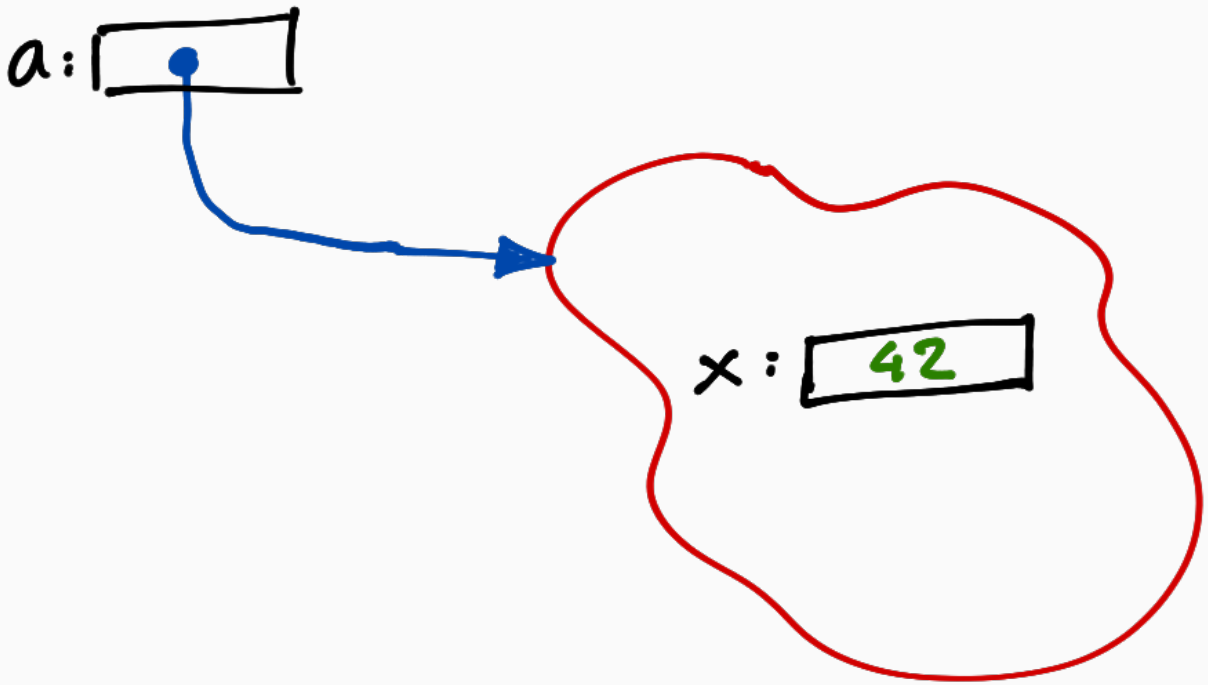
# Objetos, referencias y variables

```
public class Sesion02 {
  public static void main(String args[]) {
    A a;
    a = new A();
  }
}
```

# Objetos, referencias y variables

```
public class Sesion02 {
  public static void main(String args[]) {
    A a;
    a = new A();
  }
}
```



# *Spotify*

*There are only two hard things in Computer Science: cache invalidation and naming things.*

*Phil Karlton*

explicit, nombre, título, artísta, interprete, autor, compositor, duración, álbum, audio, sonidos, notas, imagen, valoración[1]

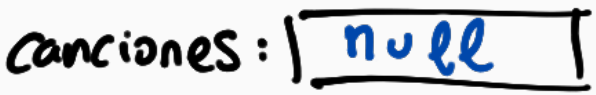
[1] Esta lista de nombres representa una votación en clase

# Punto de partida

```
class Cancion {
  String titulo;
  String interprete;
  String compositor;
  String album;
  int duracion;
  String audio;
  String imagen;
  int valoracion;
  boolean explicit;
}
```

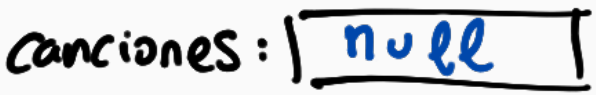
# *Lista* de reproducción

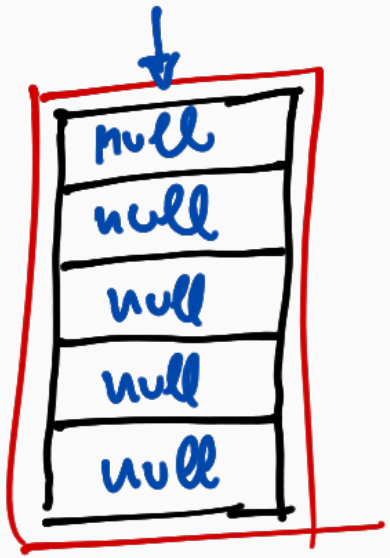
```
class Sesion03 {
  public static void main(String args[]) {
    Cancion[] canciones = new Cancion[5];
    canciones[0] = new Cancion();
  }
}
```



# *Lista* de reproducción

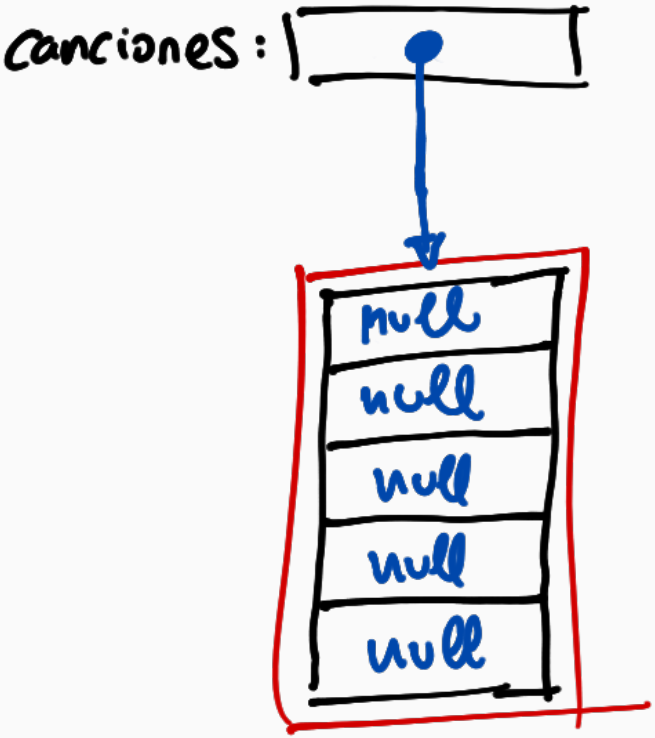
```
class Sesion03 {
  public static void main(String args[]) {
    Cancion[] canciones = new Cancion[5];
    canciones[0] = new Cancion();
  }
}
```



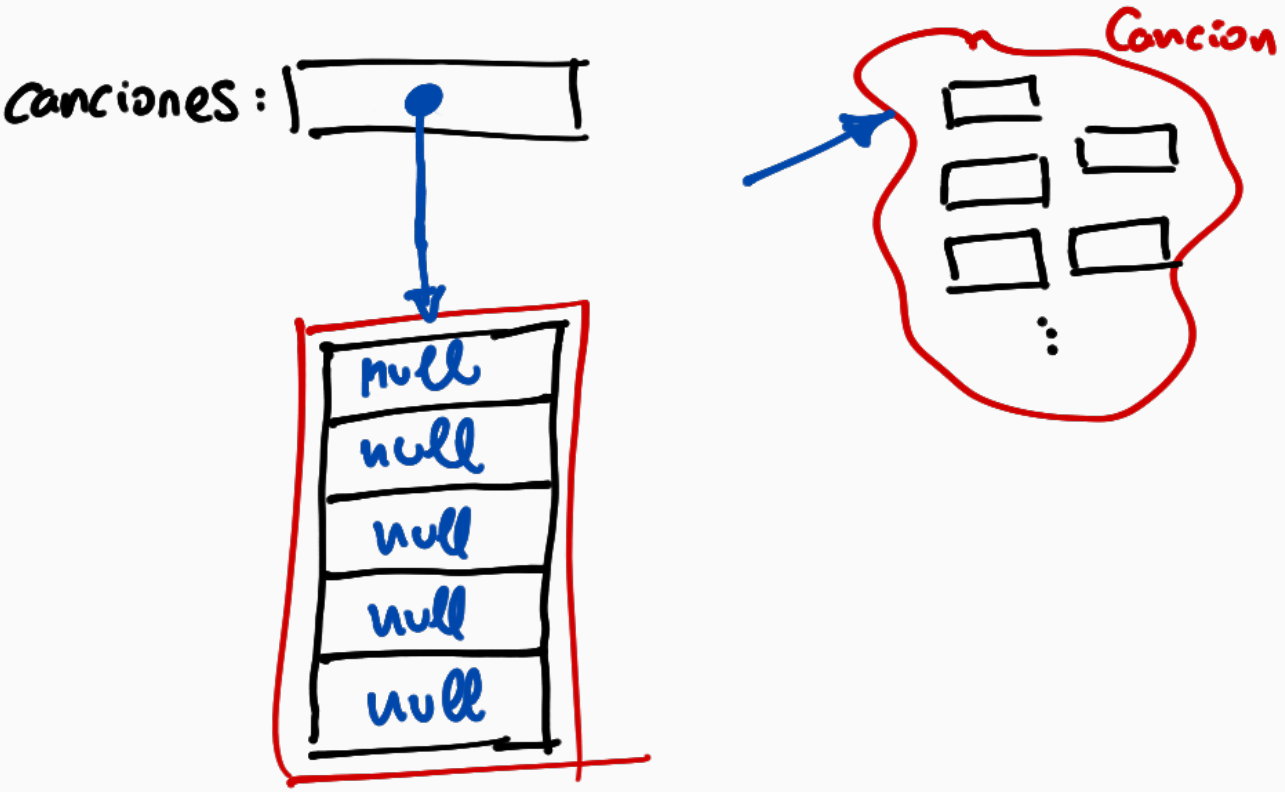
# *Lista* de reproducción

```
class Sesion03 {
  public static void main(String args[]) {
    Cancion[] canciones = new Cancion[5];
    canciones[0] = new Cancion();
  }
}
```



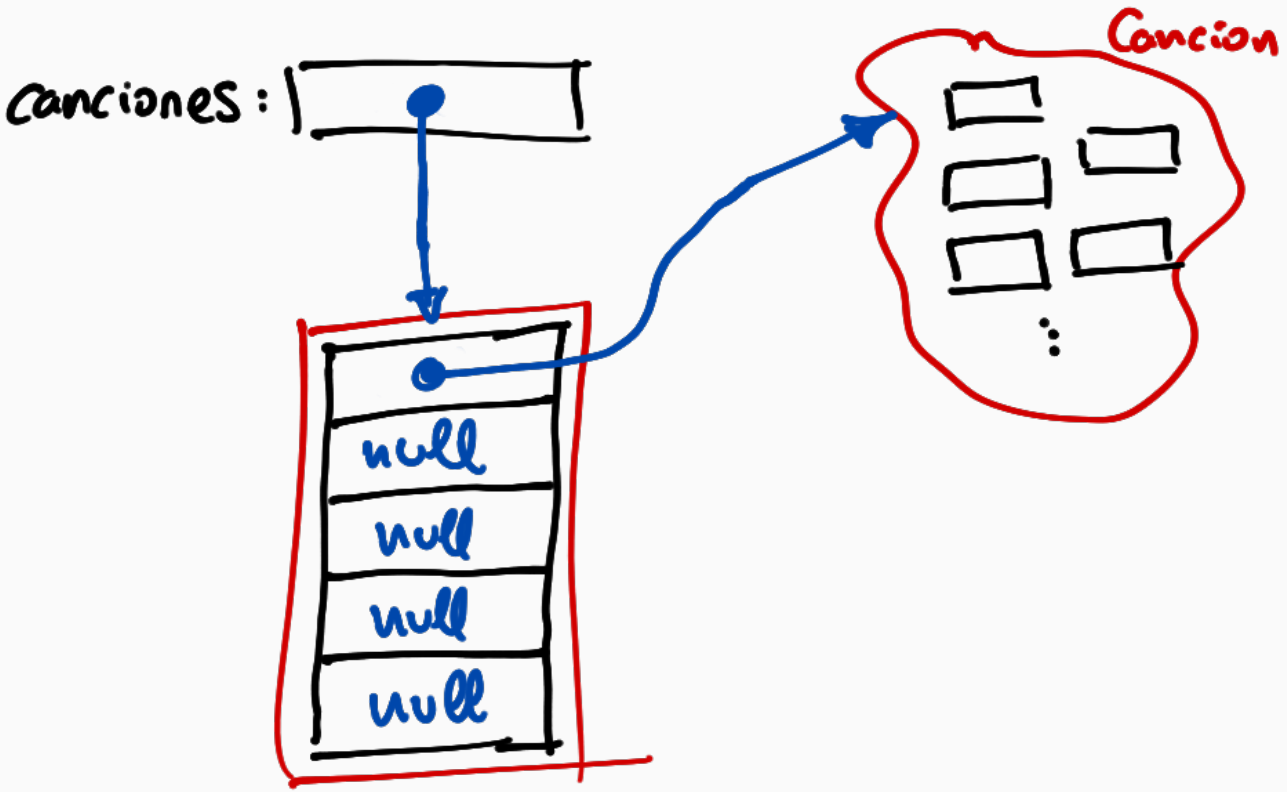
# *Lista* de reproducción

```
class Sesion03 {
  public static void main(String args[]) {
    Cancion[] canciones = new Cancion[5];
    canciones[0] = new Cancion();
  }
}
```



# *Lista* de reproducción

```
class Sesion03 {
  public static void main(String args[]) {
    Cancion[] canciones = new Cancion[5];
    canciones[0] = new Cancion();
  }
}
```



# ¡Los *arrays* de Java son objetos! :o

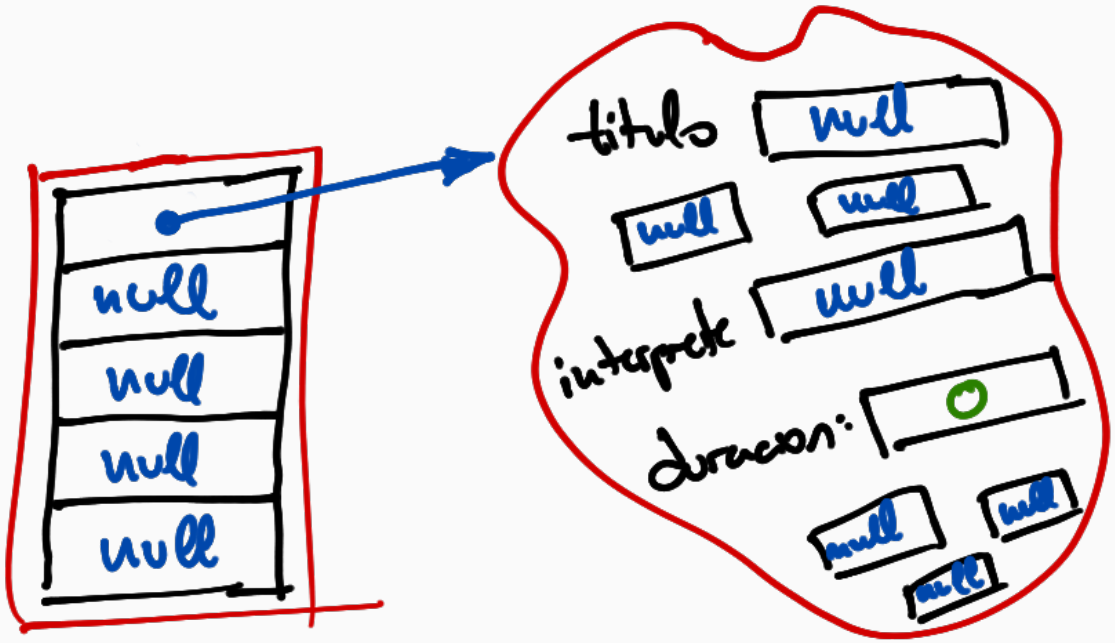
- `Cancion[]` es una clase (o tipo)
- Si $A$ es un tipo entonces $A$`[]` es un tipo
- Es el tipo de los arrays de $A$s
- Declaramos variables poniendo el tipo delante delante: `Cancion[] canciones;`
- Las **variables array son referencias**
- El objeto array **no existe hasta hacer el `new`**
- **Sus elementos son variables** inicializadas a

  **`null`**

  **`true`**, `0`, `0.0`, etc.

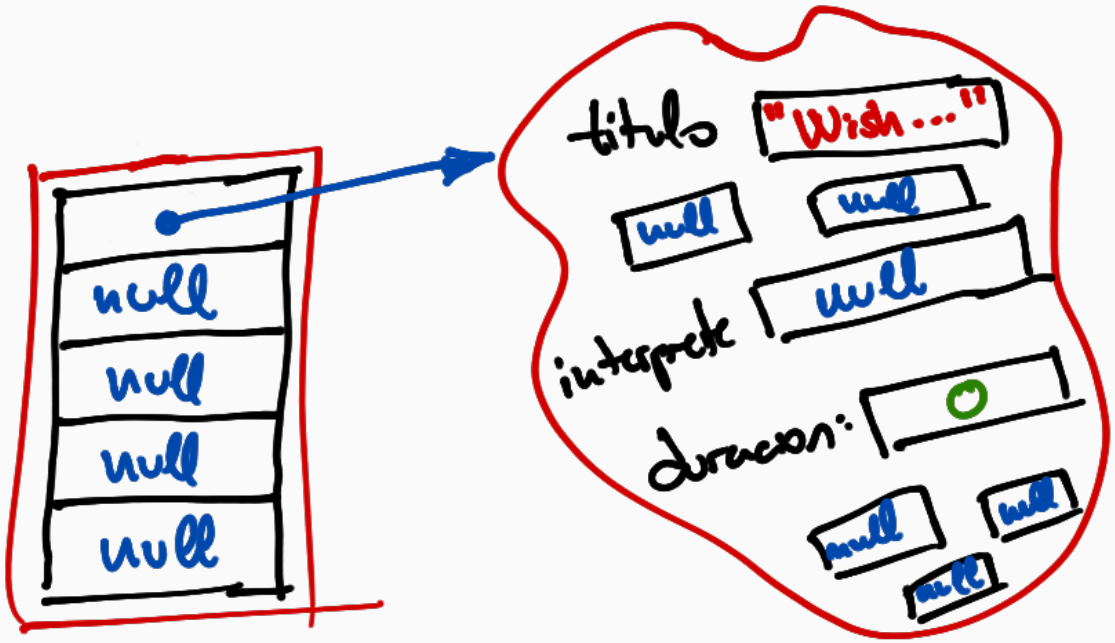
# Primera canción

```
canciones[0] = new Cancion();
canciones[0].titulo = "Wish you were here";
canciones[0].interprete = "Pink Floyd";
canciones[0].duracion = 634;
```



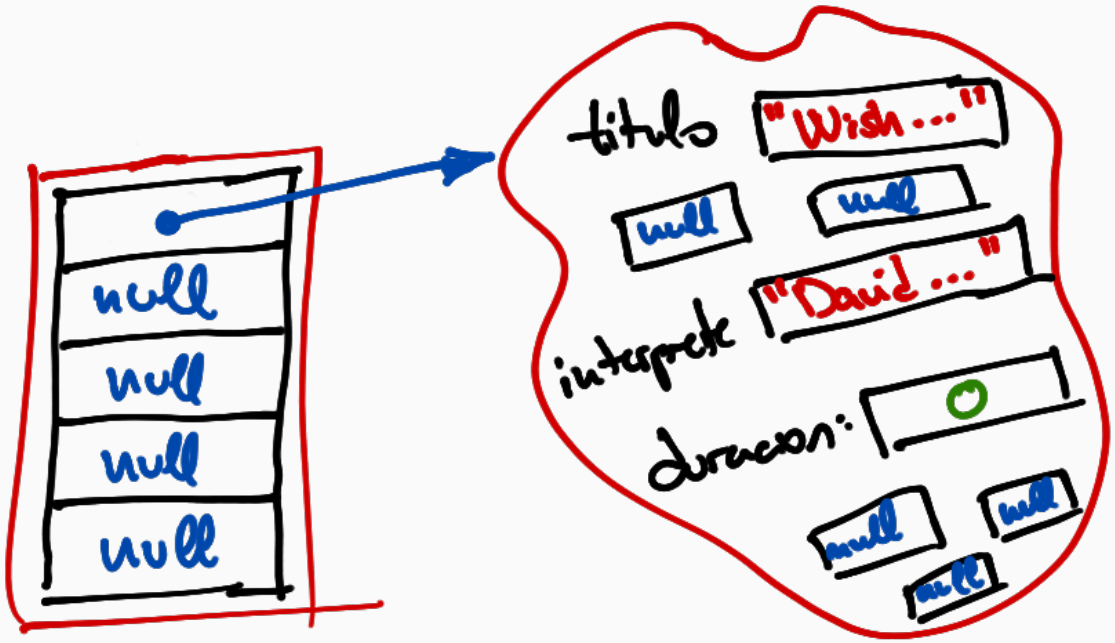
# Primera canción

```
canciones[0] = new Cancion();
canciones[0].titulo = "Wish you were here";
canciones[0].interprete = "Pink Floyd";
canciones[0].duracion = 634;
```



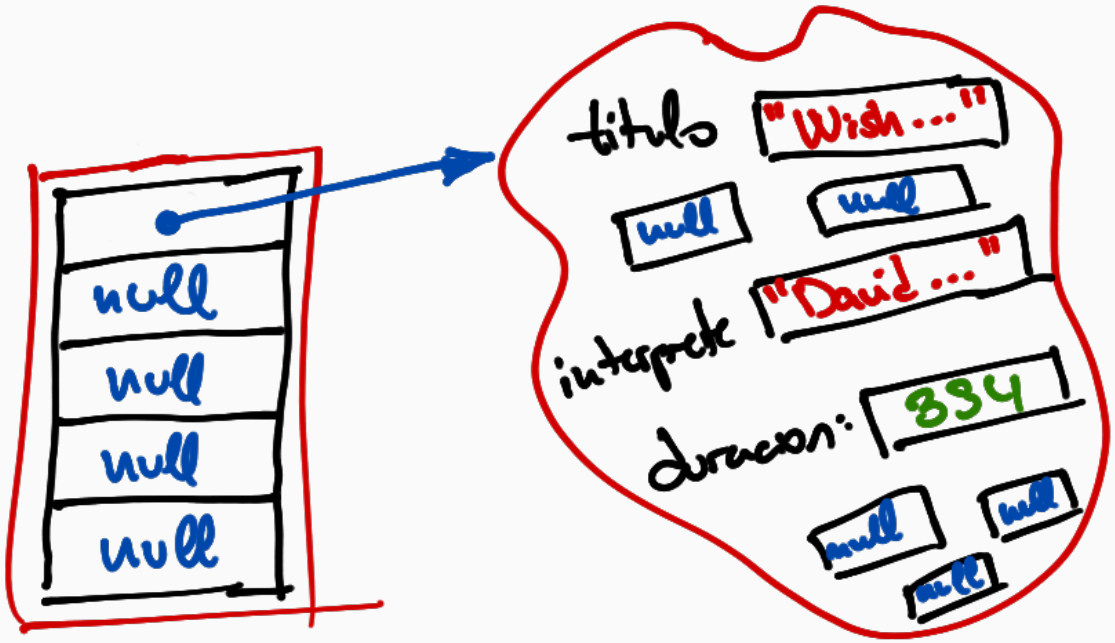
# Primera canción

```
canciones[0] = new Cancion();
canciones[0].titulo = "Wish you were here";
canciones[0].interprete = "Pink Floyd";
canciones[0].duracion = 634;
```



# Primera canción

```
canciones[0] = new Cancion();
canciones[0].titulo = "Wish you were here";
canciones[0].interprete = "Pink Floyd";
canciones[0].duracion = 634;
```



# Rellenad un par de canciones i

# Rellenad un par de canciones i

```
canciones[0] = new Cancion();
canciones[0].titulo = "Wish you were here";
canciones[0].interprete = "Pink Floyd";
canciones[0].compositor = "David Gilmour, Roger Waters";
canciones[0].album = "Wish you were here";
canciones[0].duracion = 334;
canciones[0].audio = "RocNidUrc6.mp3";
canciones[0].imagen = "RocNidUrc6.jpg";
canciones[0].valoracion = 5;
```

# Rellenad un par de canciones ii

```
canciones[1] = new Cancion();
canciones[1].titulo = "The logical song";
canciones[1].interprete = "Supertramp";
canciones[1].compositor = "Rick Davies, Roger Hodgson";
canciones[1].album = "Breakfast in America";
canciones[1].duracion = 255;
canciones[1].audio = "wrewoig0.mp3";
canciones[1].imagen = "wrewoig0.jpg";
canciones[1].valoracion = 5;
```

# Imprimir las canciones *bonitas*

```
System.out.println(
  "| " + canciones[0].titulo
  + " | " + canciones[0].interprete
  + " | " + canciones[0].duracion / 60 + "mins |"
);
System.out.println(
  "| " + canciones[1].titulo
  + " | " + canciones[1].interprete
  + " | " + canciones[1].duracion / 60 + "mins |"
);
```

---

```
| Wish you were here | Pink Floyd | 5mins |
| The logical song | Supertramp | 4mins |
```

# Todos suspensos

# Todos suspensos

Pero ya lo sabíamos. . .

*Everything should be built top-down, except the first time.*

*Epigrams on Programming (Alan J. Perlis)*

# Dos roles para ver un código

Cuando lo implementas

y

Cuando lo usas

# Cuando usas un código

- No quieres saber cómo está hecho
- Sólo quieres saber qué hace
- No quieres repetir trabajo

# Cuando usas un código

- No quieres saber cómo está hecho
- Sólo quieres saber qué hace
- No quieres repetir trabajo

Cuando programas `Sesion03.java`

# Cuando implementas un código

- Estás al servicio de quienes lo usan
- Que no necesiten mirar el cómo
- Explicar claramente el qué

# Cuando implementas un código

- Estás al servicio de quienes lo usan
- Que no necesiten mirar el cómo
- Explicar claramente el qué

Cuando programas `Cancion.java`

# Objetivo: encapsular el comportamiento

- Escribimos una forma de crear canciones
- Escribimos una forma de sacar los minutos
- Escribimos una forma de formatear las canciones

# Objetivo: encapsular el comportamiento

- Escribimos una forma de crear canciones
- Escribimos una forma de sacar los minutos
- Escribimos una forma de formatear las canciones

Encapsular

ese comportamiento

en la clase `Cancion`

(junto con los datos)

# ¿Imaginais?

```
canciones[0] =
  new Cancion("Wish you were here",
              "Pink Floyd",
              "David Gilmour, Roger Waters",
              354);
```

# ¿Imaginais?

```
System.out.println(cancion[0].minutos());
```

---

# ¿Imaginais?

```
System.out.println( cancion[0] .minutos());
```

# ¿Imaginais?

```
System.out.println(.minutos());
```

---

5mins

# ¿Imaginais incluso?

```
System.out.println(cancion[0]);
```

---

```
| Wish you were here | Pink Floyd | 5mins |
```

# ¿Imaginais?

Yes, we can