

CLEAN CODE

FORMATO

Daniel Blanco Calviño

UN BUEN FORMATO FACILITA LA LECTURA

- Formato vertical. Densidad, distancia y orden.
- Formato horizontal. Densidad e indentación.
- Recomendación personal, seguir el **styleguide de Google** -> <https://google.github.io/styleguide/javaguide.html>
- **Configurar el IDE para que aplique el formato automáticamente al guardar los cambios.**
- **Todo el equipo de trabajo debe escribir bajo las mismas reglas!**

FORMATO VERTICAL - DENSIDAD

```
public class Car {  
    private Battery battery;  
    private Engine engine;  
    public void start() {  
        battery.connect();  
        engine.start();  
    }  
    public void stop() {  
        engine.stop();  
        battery.disconnect();  
    }  
    public void checkBattery() {  
        battery.checkBatteryLife();  
    }  
    public void checkEngine() {  
        engine.start();  
    }  
}
```

```
public class Car {  
    private Battery battery;  
    private Engine engine;  
  
    public void start() {  
        battery.connect();  
        engine.start();  
    }  
  
    public void stop() {  
        engine.stop();  
        battery.disconnect();  
    }  
  
    public void checkBattery() {  
        battery.checkBatteryLife();  
    }  
  
    public void checkEngine() {  
        engine.start();  
    }  
}
```

FORMATO VERTICAL - DISTANCIA

Agrupar conceptos relacionados y separar conceptos diferentes.



```
public String getUserConfigAsString() {  
    Session session = getSession();  
  
    User currentUser = session.get(username);  
  
    String configAsString = null;  
  
    if(currentUser.isValid()) {  
        Config config = currentUser.getConfig();  
        configAsString = parseConfig(config);  
    }  
  
    return configAsString;  
}
```



```
public String getUserConfigAsString() {  
    Session session = getSession();  
    User currentUser = session.get(username);  
  
    String configAsString = null;  
    if(currentUser.isValid()) {  
        Config config = currentUser.getConfig();  
        configAsString = parseConfig(config);  
    }  
  
    return configAsString;  
}
```

FORMATO VERTICAL - ORDEN



```
public void c() {  
    //...  
}  
  
public void b() {  
    //...  
}  
  
public void a() {  
    b();  
    c();  
}
```



```
public void a() {  
    b();  
    c();  
}  
  
public void b() {  
    //...  
}  
  
public void c() {  
    //...  
}
```

FORMATO HORIZONTAL - DENSIDAD



```
public Integer cuadrupleSum(Integer a,Integer b,Integer c,Integer d){  
    if(a==null||b==null||c==null||d==null) {  
        return null;  
    }  
  
    return a+b+c+d;  
}
```



```
public Integer cuadrupleSum(Integer a, Integer b, Integer c, Integer d) {  
    if(a == null || b == null || c == null || d == null) {  
        return null;  
    }  
  
    return a + b + c + d;  
}
```

FORMATO HORIZONTAL - INDENTACIÓN

```
public Integer cuadrupleSum(Integer a,Integer b,Integer c,Integer d){  
    if(a==null||b==null||c==null||d==null) {  
        return null;  
    }  
    return a+b+c+d;  
}
```

- Seguir **siempre las mismas reglas**.
- Recomendado **dos o cuatro espacios** de indentación.
- Cuidado con el tamaño de línea! **Máximo recomendado 100 caracteres**.

CLEAN CODE

FORMATO

Daniel Blanco Calviño