



Ajuntament
de Barcelona

Best code practices & more

Laziness, Impatience and Hubris

Marzo 2019



¿Para que las buenas practicas?

- Mejor calidad del código
- Sistema mas eficiencia
- Sistemas fáciles de mantener

Tu yo del futuro será mas feliz



Principios básicos

- Don't Repeat Yourself (DRY)
- Keep It Simple Stupid (KISS)
- Refactor, Refactor, Refactor
(¿RRR?)



Ejemplos y casos prácticos



indentación consistente

```
public void Style1()
{
    if (condition)
    {
        timeToStrike();
        for (int i = 0; i < indexList.lastIndexOf(i); i++)
        {
            System.out.println("Style 1");
        }
    }
    else
    {
        abortMission();
    }
}
```



indentación consistente

```
public void Style2() {  
    if (condition) {  
        timeToStrike();  
        for (int i = 0; i < indexList.lastIndexOf(i); i++) {  
            System.out.println("Style 2");  
        }  
    } else {  
        abortMission();  
    }  
}
```



indentación consistente

```
public void Style3()
{
    if (condition)
    {
        timeToStrike();
        for (int i = 0; i < indexList.lastIndexOf(i); i++)
        {
            System.out.println("Style 13");
        }
    }
    else
    {
        abortMission();
    }
}
```



Idioma y nombres consistentes.

- Nombres, variables, funciones, constantes... Todo en ingles.
- Nombres descriptivos y no redundantes.

```
public class Plane
{
    public int Id;
    public String Name;
    public int CustomerSeats;
    public int PlaneWorkerSeats;
}
```

```
Plane plane = new Plane();
plane.Id = 1;
plane.CustomerSeats = 100;
plane.Name = "The crusher";
plane.PlaneWorkerSeats = 10;
```




Comentarios y orden de código.

Variables ☐ Constructor ☐ Getters/Setters ☐ Metodos

```
public class GameRules
{
    //Game Static Rules
    public static final int CLOCK_RATE_RULE = 1500;
    public static final int CLOCK_DELAY_RULE = 1000;
    public static final String SERVER_IP = "192.168.1.1";

    //GameDecisions
    public boolean isGameOn;
    public boolean isPlayerConnected;
    public boolean isWhatever;

    //GameComponents Objects
    public Plane plane1;
    public Plane plane2;

    //Namings
    public String name;
    public String name2;
    public String name3;
}
```



Convenciones de nombres I

Variables:

- `public String nameOfGame;`

Variables locales: (Poco común en java)

- `public String _auxiliar;`

Metodos:

- `public void rollDices(){}`

Interfaces:

- `public interface iEnumerable(){}`

Clases:

- `public class Plane{}`



Barcelona Activa, la agencia de desarrollo económico y local del Ayuntamiento de Barcelona





Convenciones de nombres II

Constantes:

- `public String GAME_RULES = "Britania rules the waves";`

Getters/Setters

```
public boolean isGameOn()  
{  
    return isGameOn;  
}
```

```
public void setGameOn(boolean isGameOn)  
{  
    this.isGameOn = isGameOn;  
}  
private boolean isGameOn;
```



Optimizar uso de memoria

- Reducir los Try Catch a lo imprescindible (no olvidarse del finally)
- Instanciar los objetos solo cuando sea necesario.
- Reutilizar variables y cálculos de ser posible.

```
double square = double a * double a;           // Optimized
double cube = double a * double a * double a;   // Non-optimized
double cube = double a * double square;         // Optimized
double quad = double a * double a * double a * double a; // Non-optimized
double quad = double square * double square;    // Optimized
```

- Cerrar siempre las conexiones a base de datos, Buffers, readers...
- Pensar bien el contenido de un bucle (por ejemplo concatenaciones)



Modularizar

- Un método no debería tener mas de una funcionalidad
- Las repeticiones de código se pueden modularizar dentro de un método.
 - Ej: ejercicio vehiculos.
- Tambien se debe modularizar a mas alto nivel, clases, packages, modelos...



Orden y limpieza del proyecto

- No saltarse los patrones de código del proyecto (MVC, Patron repository, capas...)
- Ordenar mediante carpetas incluso dentro del dominio por secciones.

ARCHIVOS	FECHA DE MODIFICACIÓN	DESCRIPCIÓN	TAMAYO
.vs	20/03/2019 22:12	Carpeta de archivos	
Delicapp2.Angular	20/03/2019 22:12	Carpeta de archivos	
Delicapp2.Lib	20/03/2019 22:12	Carpeta de archivos	
Delicapp2.Lib.Client	20/03/2019 22:12	Carpeta de archivos	
Delicapp2.MongoDbMigrator	20/03/2019 22:12	Carpeta de archivos	
Delicapp2.Server	20/03/2019 22:12	Carpeta de archivos	
Delicapp2.Server.DA.EFCore	20/03/2019 22:12	Carpeta de archivos	
Delicapp2.Unity	20/03/2019 22:12	Carpeta de archivos	
Delicapp2.Web.Client	20/03/2019 22:12	Carpeta de archivos	
Delicapp2.Web.Server	20/03/2019 22:12	Carpeta de archivos	
Delicapp2.Web.Shared	20/03/2019 22:12	Carpeta de archivos	
	20/03/2019 22:12	Documento de tex...	2 KB
Delicapp2	20/03/2019 22:12	Microsoft Visual S...	24 KB



Orden y limpieza del proyecto



