

CLEAN CODE

COMENTARIOS

Daniel Blanco Calviño

EL BUEN CÓDIGO NO NECESITA COMENTARIOS

- Añadir comentarios **solo cuando sea estrictamente necesario**.
- Son muy **complicados de mantener**. El código cambia muy rápido, y los comentarios quedan **desactualizados rápidamente**.
- Si el código tiene muchos comentarios, puede ser por dos razones:
 - El código no se entiende => Refactorización.
 - Los comentarios son obvios => Eliminar comentarios.

EJEMPLOS DE MALOS COMENTARIOS

```
//Check if password is secure  
if(password != null && password.length > 9 && !password.contains(username))
```



```
if(isSecure(password))
```



EJEMPLOS DE MALOS COMENTARIOS

```
private String date; // The date as string
```

```
private Date date; // The date as string
```

```
/*  
 * @param title The title of the Book  
 * @param description The description of the Book  
 * @param author The author of the Book  
 * @param numberOfPages The number of pages of the Book  
 *  
 * */  
public void addBook(String title, String description, String author, int numberOfPages) {
```

EJEMPLOS DE MALOS COMENTARIOS

```
➤ public void newWayOfProccessingRequests(Request request) {  
    process(request);  
}  
  
// public void oldWayOfProcessingRequests(Request request) {  
//     ...  
//     ...  
//     ...  
// }
```

Nunca dejes comentado código que ya no se está utilizando. ¡Para algo tenemos el **software de control de versiones!**

BUENOS COMENTARIOS

```
//Matches Dates in format 17-01-2021 18:03:45  
Pattern.matches("(\\d{2})-(\\d{2})-(\\d{4}) (\\d{2}):(\\d{2}):(\\d{2})", date)
```

```
//TODO Expect this method to change when the client changes auth API  
public void login(User user) {
```

Los Javadocs de una API pública son otro claro ejemplo de buenos comentarios.

CLEAN CODE

COMENTARIOS

Daniel Blanco Calviño