CLEAN CODE

NOMBRES SIGNIFICATIVOS

Daniel Blanco Calviño

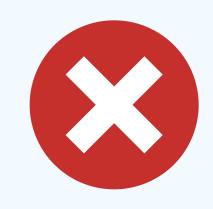
DEMASIADOS NOMBRES...

- Variables, funciones, argumentos.
- Clases, paquetes.
- Ficheros y directorios.
- Ficheros de despliegue.



NOMBRES QUE REVELEN INTENCIÓN

```
int d;
int m;
int y;
```



```
int d; // Day
int m; // Month
int y; // Year
```

```
int dayOfBirth;
int monthOfBirth;
int yearOfBirth;
```



```
for (int i = 0; i < numberOfRequests; i++) {
   processRequest(requests.get(i));
}</pre>
```

EVITA LA DESINFORMACIÓN

¿Encuentras la diferencia?

```
public class ServiceForEfficientUpdateMysqlDatabase {
public class ServiceForEfficientBackupMysqlDatabase {
```

```
¿ Es realmente una lista? ¿ No son lo mismo?
```

```
Map<Integer, String> employeeList;
```

```
private class ProductData {}
private class ProductInfo {}
```

USA NOMBRES PRONUNCIABLES

Los nombres deben ser fáciles de buscar

```
public class InfRcrd {
    private String nm;
    private String snm;
}
public class Employee {
    private String name;
    private String snm;
}
```

Es mucho mejor elegir un nombre muy largo a uno que no deja claro su significado

private String lstUsedName;





private String leastUsedName;

NOMBRES DE CLASES Y MÉTODOS

- Los nombres de las clases deben ser un nombre o conjunto de nombres, no deberían ser verbos.
- Los nombres de los métodos deberían ser verbos, indicando una acción.
- No llamar a una clase CreateEmployee, ni a un método EmployeeName(), por ejemplo.

```
public class DateParser {
    public Date parse(String date);
}
```

AÑADE CONTEXTO QUE APORTE SIGNIFICADO

- Variables firstName, lastName, street, houseNumber, city, state, zipcode.
- Estando juntas, es obvio que se trata de una dirección.
- Añadir prefijo addr para aportar información.
- user.getFirstName() vs user.getAddrFirstName().

CLEAN CODE

NOMBRES SIGNIFICATIVOS

Daniel Blanco Calviño