

# Refactoring

**Mejorando el diseño del código existente**

# ¿Refactoring? ¿Qué es?



# ¿Refactoring? ¿Qué es?

## Refactoring:

**Una modificación hecha en la estructura interna de una aplicación de modo tal que sea más fácil de entender y más simple de modificar, sin cambiar su funcionamiento.**

# ¿Refactoring? ¿Qué es?

## Frase épica:

**Cualquier *salame* puede escribir código que una computadora pueda entender. Los buenos programadores escriben código que otros programadores pueden entender.**

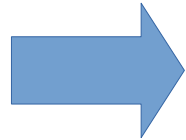
**Martin Fowler**

# ¿Para qué Refactorizamos?

- Simplificar el Código
- Remover Código Duplicado
- Mejorar el Diseño
  - Transformar algo procedural en OO
  - Reducir acoplamiento
  - Mejorar Cohesión
  - Separar en capas
- Hacerlo mas expresivo/legible (nombrar el código)
- Remover código muerto
- Todo esto puedo hacerse con seguridad únicamente si tengo Testing Automatizado.

# Refactoring: Ejemplo

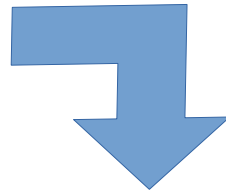
```
class Concurso {  
    ...  
    public boolean participanteInscripto(  
        Participante participante) {  
        boolean esta = false;  
        int i = 0;  
        List<Inscripcion> l = this.inscriptos;  
        while (!esta && i < l.size()) {  
            if (l.get(i).equals(participante)) {  
                esta = true;  
            } else {  
                i++;  
            }  
        }  
        return esta;  
    }  
}
```



```
class Concurso {  
    ...  
    public boolean estaInscripto(  
        Participante participante) {  
        for (Participante p : this.inscriptos) {  
            if (p.equals(participante)) {  
                return true;  
            }  
        }  
        return false;  
    }  
}
```

# Refactoring: Aún mejor...

```
class Concurso {  
    ...  
    public boolean estaInscripto(  
        Participante participante) {  
        for (Participante p : this.inscriptos) {  
            if (p.equals(participante)) {  
                return true;  
            }  
        }  
        return false;  
    }  
}
```



```
class Concurso {  
    ...  
    public boolean estaInscripto(Participante participante) {  
        boolean estaInscripto = this.inscriptos.stream()  
            .anyMatch(p -> participante.equals(p));  
        return estaInscripto;  
    }  
}
```

# Refactoring++

- **Catálogo de Refactoring**
  - <https://refactoring.com/catalog> (revisemos algunas)
- **Clean Code**
  - Fácil de modificar
  - Entendible por otros programadores
  - Sin código duplicado
  - Con testing automatizado
- **Code Smell**
  - <https://sourcemaking.com/refactoring/smells>
- **Technical Debt** (Ward Cunningham)



# Bibliografía

- **Refactoring, Improving the Design of Existing Code.** Martin Fowler, Addison-Wesley, Professional, 1999.

