Hexacta Labs

Clean Code

PUESTO Nº6

NIVEL 3

EMPRESA INFORMÁTICA 2009

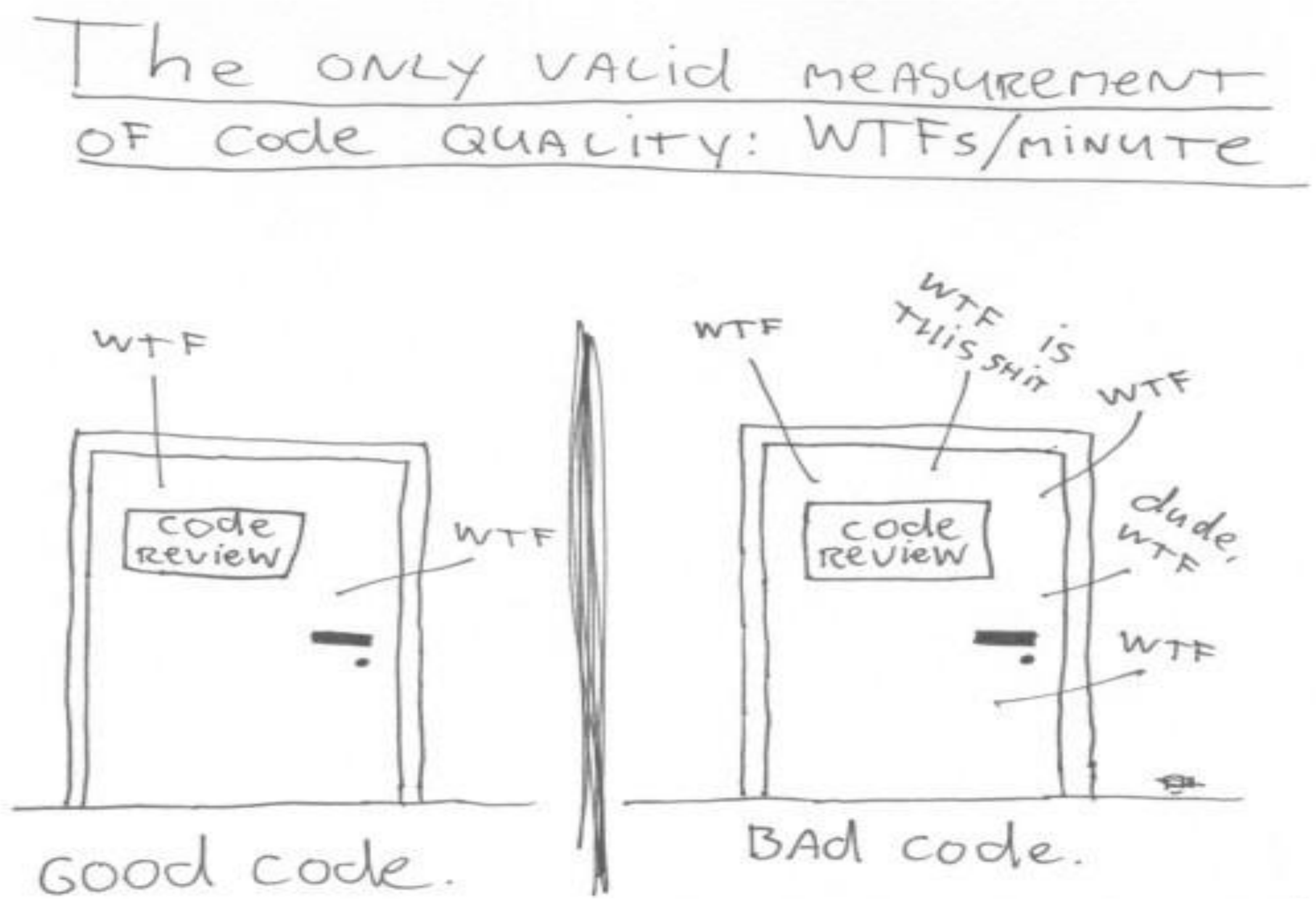
EXPORTACIÓN DE SERVICIOS

Agenda

- ¿Clean code?
- Reglas básicas
- Code Smells
- Refactorings mas utilizados
- Herramientas
- Bibliografía

No imprima este documento si no es necesario. Protejamos el medio ambiente.

Índice WTF


The only valid measurement of code quality: WTFs/minute
WTF
WTF
code review
Good code.
WTF
WTF is this shit
WTF
code review
dude, WTF
WTF
BAd code.

¿ Clean Code ?

Expresivo

una sola cosa

y bien

Simple

colaboradores explícitos

0% Duplicados

No imprima este documento si no es necesario. Protejamos el medio ambiente.

somos @autores

```
import java.util.List;
/**
 * @author pepe
 */
public class CycloDetector {
    ...
    ...
    ...
}
```

~10:1

Nombres que revelen intención

```
int h; //hours since game started

int hoursSinceGameStarted;
```

Nombres que revelen intención

```
boolean linearSearchFor(Object element)
```

```
boolean includes(Object element)
```

Nombres pronunciables

```
public DateFormatContainer(String dfStr) {
    this.dfStr = dfStr;
}
```

```
public DateFormatContainer(String aDateFormat) {
    this.dateFormat = aDateFormat;
}
```

Único nivel de abstracción

```
public void …() {
      Html html = new Html();
      html.addText("Hola");
      StringBuffer buffer = new StringBuffer()
      buffer.append("<p>holaaaa</p></br>");
      buffer.append("<p>como les va!</p>");
      html.addFragment(buffer.toString());
}
```

No imprima este documento si no es necesario. Protejamos el medio ambiente.

Único nivel de abstracción

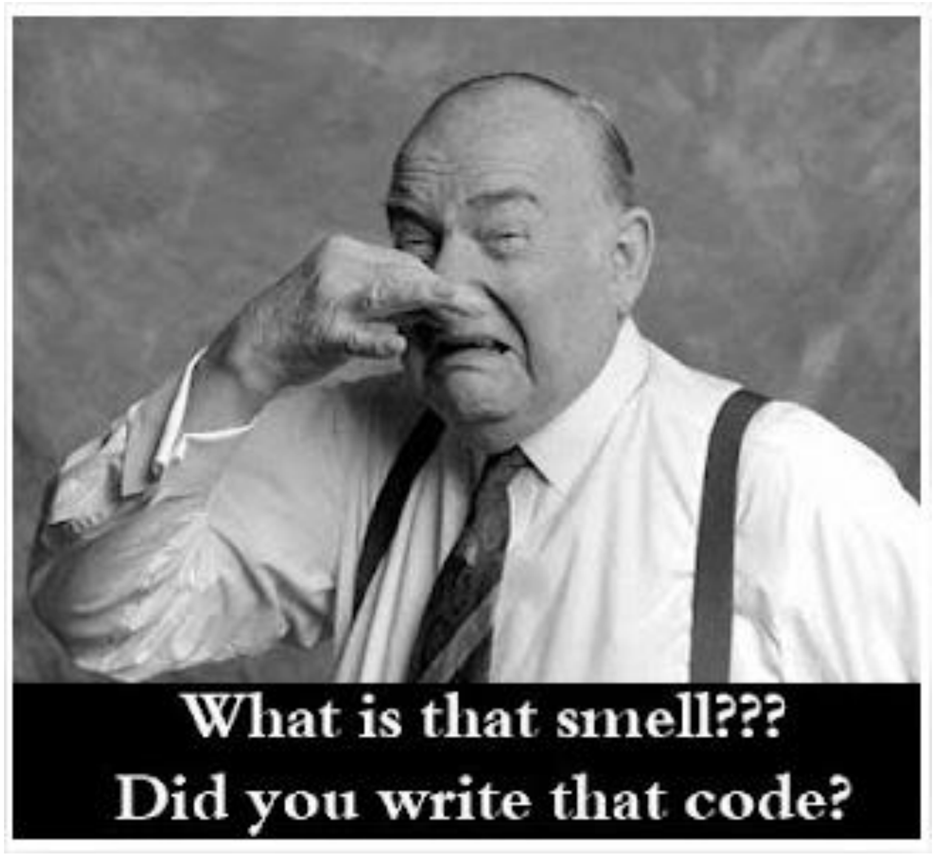
```
public void …() {
      Html html = new Html();
      html.addText("Hola");
      html.addParagraph("holaaaa");
      html.addBreak();
      html.addParagraph("como les va!“);
}
```

Clean Code – Reglas básicas

Sin efectos secundarios

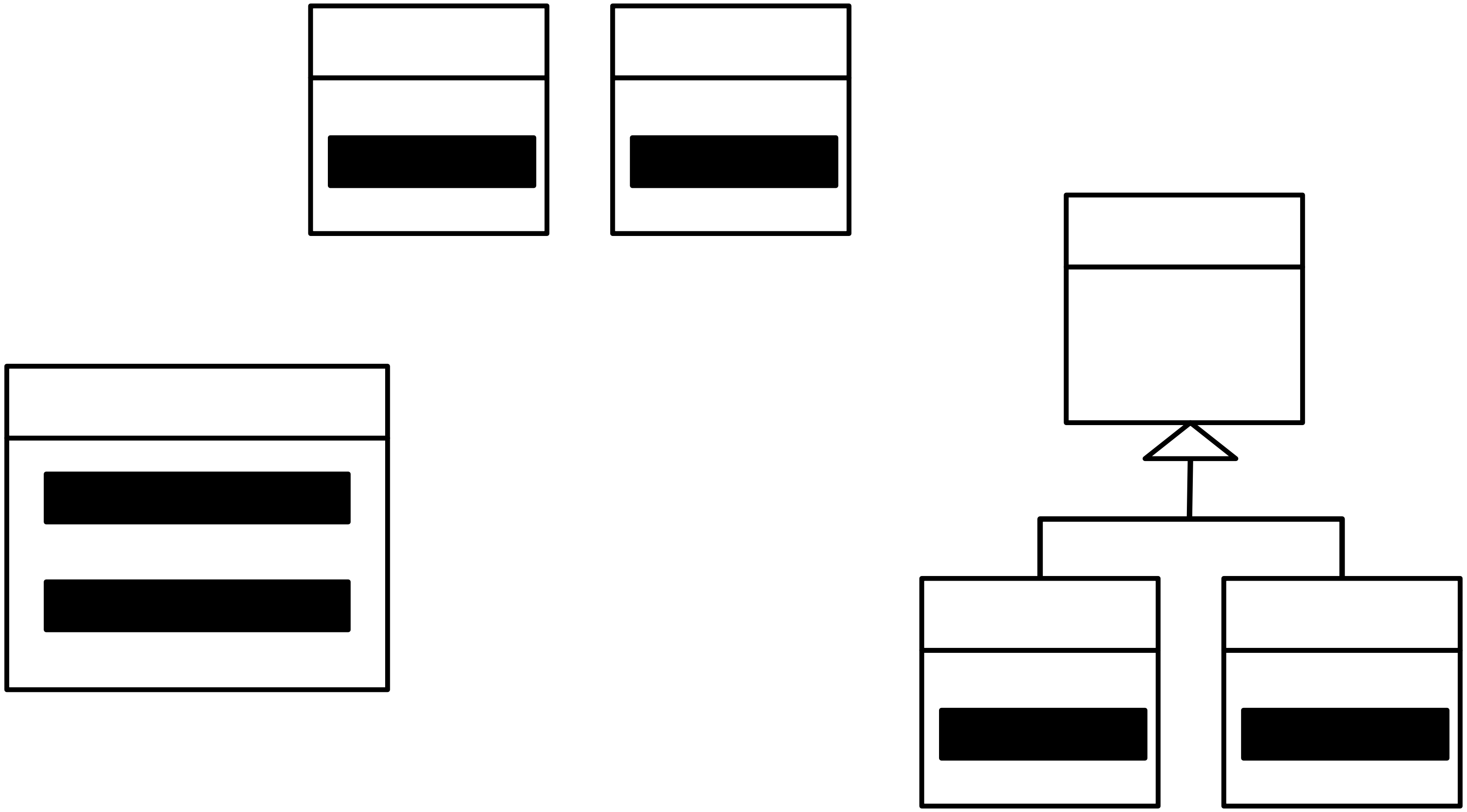
```
public boolean checkPassword(User user, String password) {
  Phrase userCodedPhrase = user.getPhraseEncodedByPassword();
  Phrase phrase = cryptographer.decryp(password);

  if (phrase.sameAs(userCodedPhrase)) {
    Session.initialize();
    return true;
  }

  return false;
}
```

Code smells



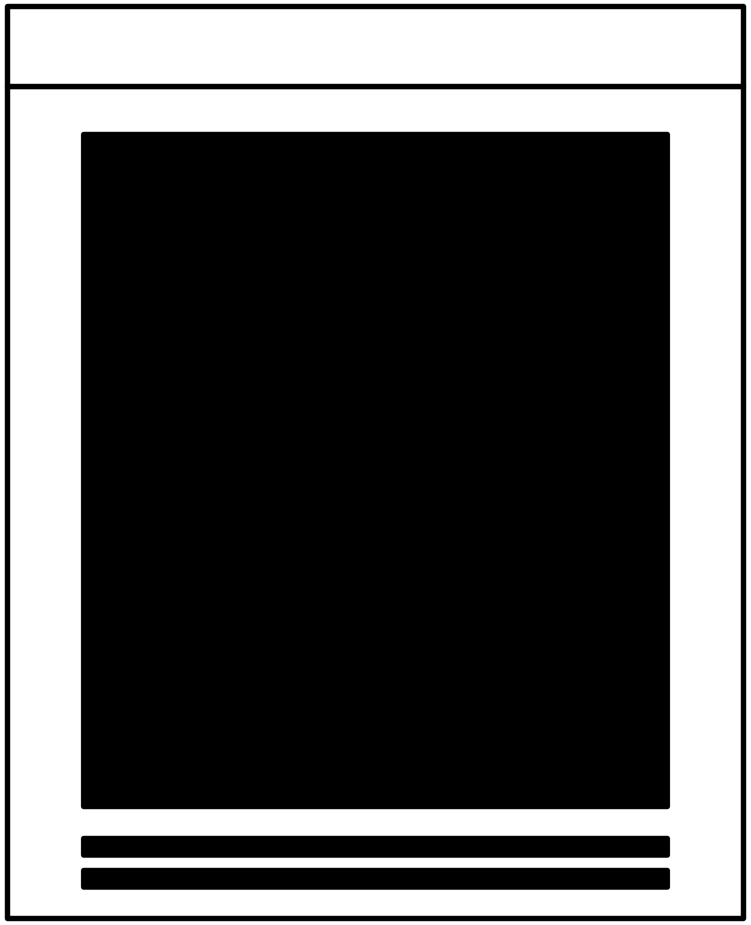
No imprima este documento si no es necesario. Protejamos el medio ambiente.

Duplicated Code

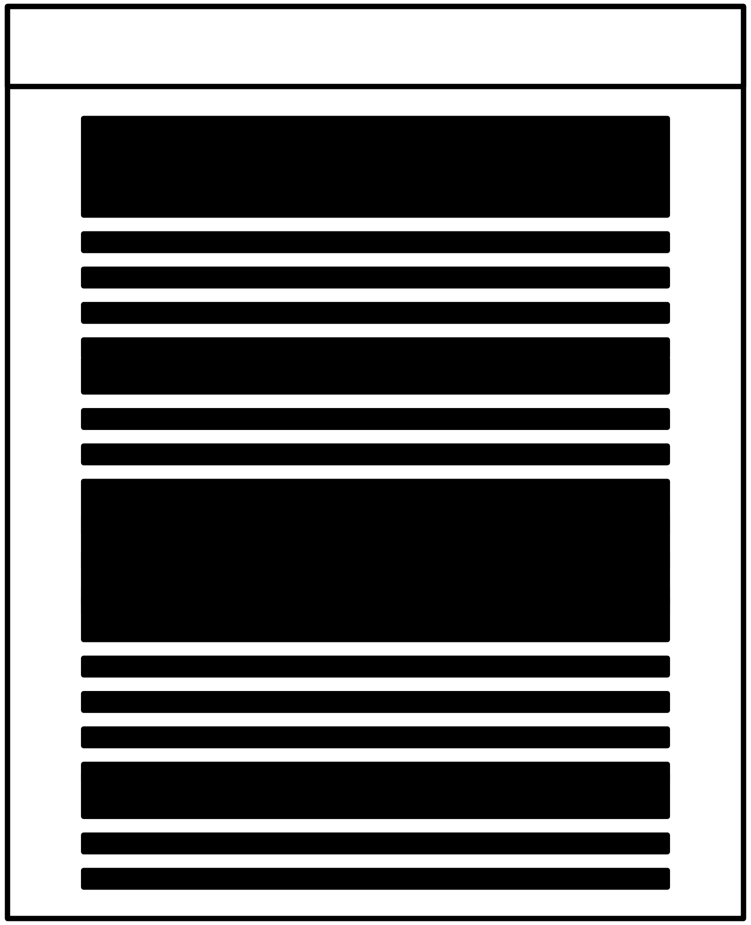


No imprima este documento si no es necesario. Protejamos el medio ambiente.

Long Method



Large Class



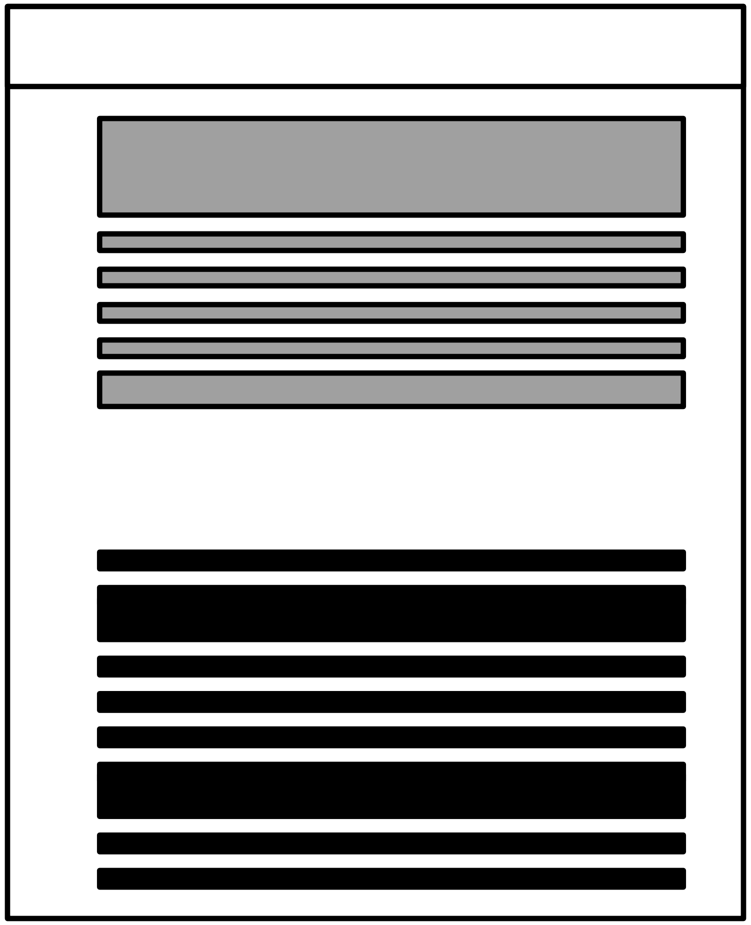
No imprima este documento si no es necesario. Protejamos el medio ambiente.

Long parameter list

xxxxxxx (■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■)

No imprima este documento si no es necesario. Protejamos el medio ambiente.

Divergent Change



No imprima este documento si no es necesario. Protejamos el medio ambiente.

feature envy

No imprima este documento si no es necesario. Protejamos el medio ambiente.

```
class CapitalCalculator {
 ...

 public double capital(Loan loan) {
  if (loan.getExpiry() == null && loan.getMaturity() != null)
   return loan.getCommitment()*loan.duration()*loan.riskFactor();

  if (loan.getExpiry() != null && loan.getMaturity() == null) {
    if (loan.getUnusedPercentage() != 1.0)
       return loan.getCommitment() * loan.getUnusedPercentage() *
                 loan.duration() * loan.riskFactor();
    else
       return (loan.outstandingRiskAmount()*loan.duration()
                   * loan.riskFactor())
              + (loan.unusedRiskAmount() * loan.duration()
                   * loan.unusedRiskFactor());
   }

   return 0.0;
 }
 ...
}
```

Data Clumps

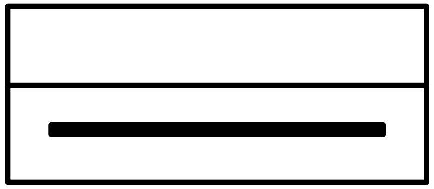
method1(■ ▲ ◆ ●)

method2(■ ▲ ◆ ●)

method3(■ ▲ ◆ ●)

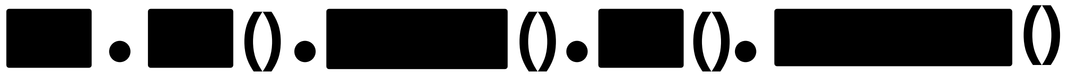
method4(■ ▲ ◆ ●)

Lazy Class

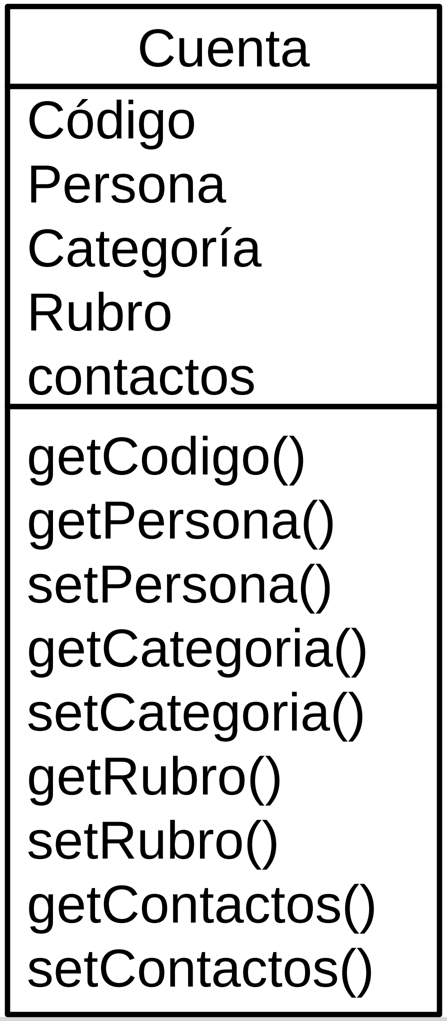


No imprima este documento si no es necesario. Protejamos el medio ambiente.

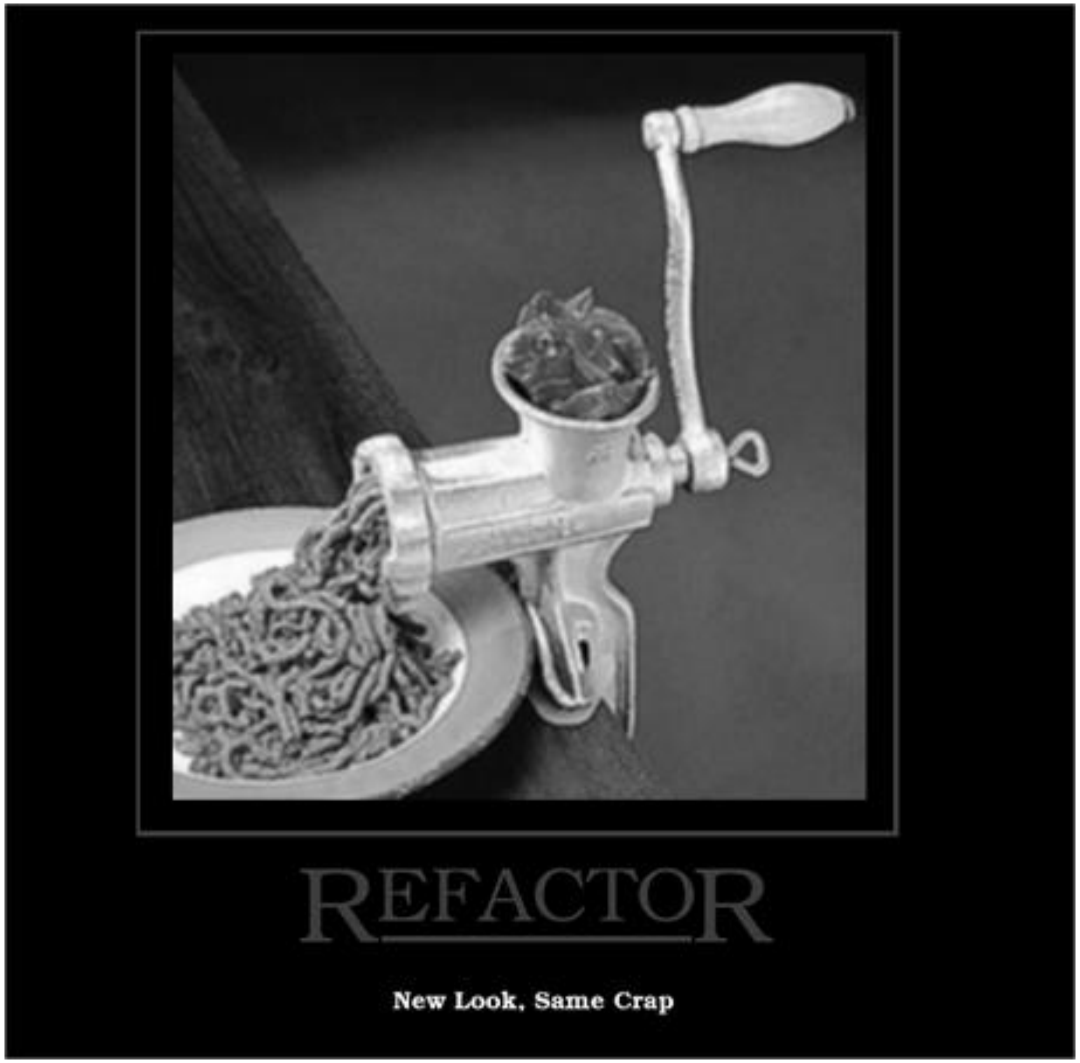
Message Chains



Data Class


Cuenta
Código
Persona
Categoría
Rubro
contactos
getCodigo()
getPersona()
setPersona()
getCategoria()
setCategoria()
getRubro()
setRubro()
getContactos()
setContactos()

Refactorings mas utilizados



No imprima este documento si no es necesario. Protejamos el medio ambiente.

1 38055 65154 7

```
if ((platform.toUpperCase().indexOf("MAC") > -1)
     && (platform.toUpperCase().indexOf("IE") > -1)
     && wasInitialized()
     && resize > 0) {

   someCode();

}

otherCode();
```

Introduce Explaining Method

```
if (isPlatformSupported()
        && wasInitialized()
        && wasResized()) {

    someCode();
}

otherCode();
```

```
boolean wasResized() {
   return resize > 0;
}

boolean isIEBrowser() {
   return platform.toUpperCase().indexOf("IE") > -1;
}

boolean isMacOs() {
   return platform.toUpperCase().indexOf("MAC") > -1;
}

boolean isPlatformSupported() {
   return isMacOs() && isIEBrowser();
}
```

1 38055 65154 7

No imprima este documento si no es necesario. Protejamos el medio ambiente.

```
double getDistanceTravelled (int time) {
    double result;
    double acc = primaryForce / mass;
    int primaryTime = Math.min(time, delay);
    result = 0.5 * acc * primaryTime * primaryTime;
    int secondaryTime = time - delay;

    if (secondaryTime > 0) {
        double primaryVel = acc * delay;
        acc = (primaryForce + secondaryForce) / mass;
        result += primaryVel * secondaryTime + 0.5 * acc *
secondaryTime * secondaryTime;
    }

    return result;
}
```

Split Temporary Variable

```
double getDistanceTravelled(int time) {
   double result;

   double primaryAcc = primaryForce / mass;
   int primaryTime = Math.min(time, delay);
   result = 0.5 * primaryAcc * primaryTime * primaryTime;
   int secondaryTime = time - delay;

   if (secondaryTime > 0) {
      double primaryVel = primaryAcc * delay;
      double secondaryAcc = (primaryForce + secondaryForce) / mass;
      result += primaryVel * secondaryTime + 0.5
              * secondaryAcc * secondaryTime * secondaryTime;
   }

   return result;
}
```

1 38055 65154 7

```
int discount(int value, int quantity) {
        if (value > 50) {
          value -= 2;
        }
        if (quantity > 100) {
          value -= 5;
        }
        return value;
}
```

Remove assignments to parameters

No imprima este documento si no es necesario. Protejamos el medio ambiente.

```
int discount(final int value, final int quantity) {
        int discount = value;
        if (value > 50) {
            discount -= 2;
        }
        if (quantity > 100) {
            discount -= 5;
        }
        return discount;
    }
```

1 38055 65154 7

```
class Page {
      private String[] lines;

      private double widthNumber;
      private String widthUnits;

      private double heightNumber;
      private String heightUnits;

      /**
       * return the page area in inches.
       */
      public double area() {
         double widthInches;
         double heightInches;
         widthInches = widthNumber *
                         ((widthUnits.equals("mm")) ? 25.4 : 1.0);
         heightInches = heightNumber *
                         ((heightUnits.equals("mm")) ? 25.4 : 1.0);
         return widthInches * heightInches;
      }

      ...
   }
```

Extract Class

```
class Length {
      private final double magnitude;
      private final Unit unit;

      public Length(Unit unit, double magnitude) {
         this.unit = unit;
         this.magnitude = magnitude;
      }

      private static Length newInInches(double magnitudeInInches) {
         return new Length(Unit.inches, magnitudeInInches);
      }

      public Length multipliedBy(Length aLength) {
         return Length.newInInches(this.magnitudeInInches()
                             + aLength.magnitudeInInches());
      }

      private double magnitudeInInches() {
         return magnitude;
      }

      private double magnitudeInMM() {
         return magnitude * Unit.mmFactor();
      }
}
```

No imprima este documento si no es necesario. Protejamos el medio ambiente.

```
class Page {
      private String[] lines;

      private Length width;
      private Length height;

      public Length area() {
         return width.multipliedBy(height);
      }
}
```

No imprima este documento si no es necesario. Protejamos el medio ambiente.

1 38055 65154 7

```
double chargeFor(Date date, int quantity) {
      double totalCharge = 0;
      if (date.after(WINTER_START) && date.before(WINTER_END)) {
         totalCharge = quantity * WINTER_RATE
                                 + WINTER_SERVICE_CHARGE;
      } else {
         totalCharge = quantity * NORMAL_RATE;
      }
      return totalCharge;
}
```

```
      if !(date.before(WINTER_START) || date.after(WINTER_END)){
```

No imprima este documento si no es necesario. Protejamos el medio ambiente.

Decompose Conditional

```
double chargeFor(Date date, int quantity) {
      if (isAWinter(date)) {
         return winterCharge(quantity);
      }
      return normalCharge(quantity);
}
```

```
double chargeFor(Date date, int quantity) {
      return isAWinter(date) ?
            winterCharge(quantity) : normalCharge(quantity);
}
```

No imprima este documento si no es necesario. Protejamos el medio ambiente.

```
private boolean isAWinter(Date date) {
    return date.after(WINTER_START) || date.before(WINTER_END);
}

private double normalCharge(int quantity) {
    return quantity * NORMAL_RATE;
}

private double winterCharge(int quantity) {
    return quantity * WINTER_RATE + WINTER_SERVICE_CHARGE;
}
```

No imprima este documento si no es necesario. Protejamos el medio ambiente.

1 38055 65154 7

No imprima este documento si no es necesario. Protejamos el medio ambiente.

```
class TicTacToeGame {

        boolean isGameOver() {
           if (allPositionsAreFilled()) {
              return true;
           }
           if (oneRowIsFilledByOnePlayer()) {
              return true;
           }
           if (oneColumnIsFilledByOnePlayer()) {
              return true;
           }
           if (oneDiagonalIsFilledByOnePlayer()) {
              return true;
           }
           return false;
        }

}
```

Consolidate conditional expression

```
boolean isGameOver() {
    if (allPositionsAreFilled()
            || oneRowIsFilledByOnePlayer()
            || oneColumnIsFilledByOnePlayer()
            || oneDiagonalIsFilledByOnePlayer()) {
        return true;
    }
    return false;
}


boolean isGameOver() {
    return allPositionsAreFilled()
            || oneRowIsFilledByOnePlayer()
            || oneColumnIsFilledByOnePlayer()
            || oneDiagonalIsFilledByOnePlayer();
}
```

```java
public double getRate() {
    if (onVacation()) {
        if (lengthOfService() > 10) {
            return 1;
        }
    }
    return 0.5;
}
```

```java
public double getRate() {
    if (onVacation() && lengthOfService() > 10) {
        return 1;
    }
    return 0.5;
}

public double getRate() {
    return (onVacation() && lengthOfService() > 10) ? 1 : 0.5;
}
```

1 38055 65154 7

No imprima este documento si no es necesario. Protejamos el medio ambiente.

```java
public double finalPrice(double price) {
      double total = 0;
      if (isSpecialDeal()) {
         total = price * 0.95;
         changed();
      } else {
         total = price;
         changed();
      }
      return total;
}
```

No imprima este documento si no es necesario. Protejamos el medio ambiente.

Consolidate duplicate conditional fragments

No imprima este documento si no es necesario. Protejamos el medio ambiente.

```
public double finalPrice(double price) {
      double total = 0;
      if (isSpecialDeal()) {
         total = price * 0.95;
      } else {
         total = price;
      }
      changed();
      return total;
}
```

No imprima este documento si no es necesario. Protejamos el medio ambiente.

1 38055 65154 7

```
public boolean exist(String nameToFind) {
      boolean found = false;
      for (String name : names) {
         if (name.equals(nameToFind)) {
            found = true;
         }
      }
      return found;
}
```

Remove control flag

```java
public boolean exist(String nameToFind) {
        for (String name : names) {
            if (name.equals(nameToFind)) {
                return true;
            }
        }
        return false;
}
```

1 38055 65154 7

```
double getPayAmount() {
    double result;
    if (isDead) {
        result = deadAmount();
    } else {
        if (isSeparated) {
            result = separatedAmount();
        } else {
            if (isRetired) {
                result = retiredAmount();
            } else {
                result = normalAmount();
            }
        }
    }
    return result;
}
```

No imprima este documento si no es necesario. Protejamos el medio ambiente.

Replace nested conditionals with guard clauses

```
double getPayAmount() {
        if (isDead) {
            return deadAmount();
        }
        if (isSeparated) {
            return separatedAmount();
        }
        if (isRetired) {
            return retiredAmount();
        }
        return normalAmount();
}
```

1 38055 65154 7

No imprima este documento si no es necesario. Protejamos el medio ambiente.

```java
calculateWeeklyPay(true);

calculateWeeklyPay(false);
```

```java
public int calculateWeeklyPay(final boolean overtime) {
    int straightTime = Math.min(400, getHoursWorked());
    int straightPay = straightTime * getHoursRate();
    int overTime = Math.max(0, getHoursWorked() - straightTime);
    double overtimeRate = overtime ? 1.5 : 1.0 * getHoursRate();
    int overtimePay = (int) Math.round(overTime * overtimeRate);
    return straightPay + overtimePay;
}
```

Replace parameter with explicit methods

```java
public int straightPay() {
      ...
      ...
}

public int overtimePay() {
      ...
      ...
}
```

1 38055 65154 7

No imprima este documento si no es necesario. Protejamos el medio ambiente.

```java
interface ClaimsRepository {
      List<Claim> claimsReceivedIn(Date start, Date end);
      List<Claim> claimsApprovedIn(Date start, Date end);
      List<Claim> claimsRejectedIn(Date start, Date end);
}
```

Introduce parameter object

```
interface ClaimsRepository {

      List<Claim> claimsReceivedIn(Range<Date> range);

      List<Claim> claimsApprovedIn(Range<Date> range);

      List<Claim> claimsRejectedIn(Range<Date> range);

}
```

No imprima este documento si no es necesario. Protejamos el medio ambiente.

```
class RepositorioDeClientes {

      public void agregar(long id, String doc, String cuit,
              String nombre, String apellido, String telefono,
              String mail, String direccion, String localidad,
              String piso, String provincia) {

              // Agrega un nuevo cliente a la DB
      }

      public void modificar(long id, String doc, String cuit,
              String nombre, String apellido, String telefono,
              String mail, String direccion, String localidad,
              String piso, String provincia) {

              // Agrega un nuevo cliente a la DB
      }

}
```

No imprima este documento si no es necesario. Protejamos el medio ambiente.

```
class RepositorioDeClientes {

      public void agregar(Cliente cliente) {
         // Agrega un nuevo cliente a la DB
      }

      public void modificar(Cliente cliente) {
         // modifica un cliente de la DB
      }

}
```

1 38055 65154 7

No imprima este documento si no es necesario. Protejamos el medio ambiente.

```java
int withdraw(double amount) {
    if (amount > balance) {
        return -1;
    }

    balance -= amount;
    return 0;
}


void usoEnCodigoCliente () {
    if (withdraw(200) < 0) {
        handleError();
    }
    moreCode();
}
```

No imprima este documento si no es necesario. Protejamos el medio ambiente.

Replace error code with exception

```java
void withdraw(double amount) {
    if (amount > balance) {
        throw new BalanceException(balance, amount);
    }
    balance -= amount;
}


public void usoEnCodigoCliente() {
    try {
        withdraw(200);
    } catch (BalanceException e) {
        handleError();
    }
    moreCode();
}
```

1 38055 65154 7

```
class ResourcePool {

    Stack<Resource> available;
    Stack<Resource> allocated;

    Resource getResource() {
        Resource result;
        try {
            result = available.pop();
            allocated.push(result);
            return result;
        } catch (EmptyStackException e) {
            result = new Resource();
            allocated.push(result);
            return result;
        }
    }

}
```

No imprima este documento si no es necesario. Protejamos el medio ambiente.

Replace exception with test

No imprima este documento si no es necesario. Protejamos el medio ambiente.

```
class ResourcePool {

    Stack<Resource> available;
    Stack<Resource> allocated;

    Resource getResource() {
        Resource result;
        if (available.isEmpty()) {
            result = new Resource();
        } else {
            result = available.pop();
        }
        allocated.push(result);
        return result;
    }

}
```

No imprima este documento si no es necesario. Protejamos el medio ambiente.

Herramientas

No imprima este documento si no es necesario. Protejamos el medio ambiente.

Herramientas de chequeo automático

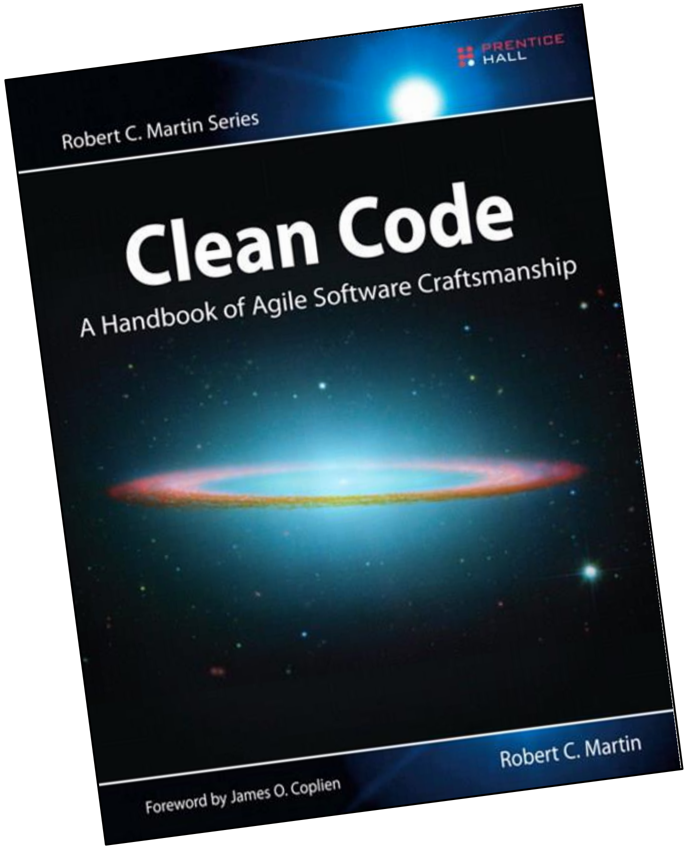
pmd
DON'T SHOOT THE MESSENGER

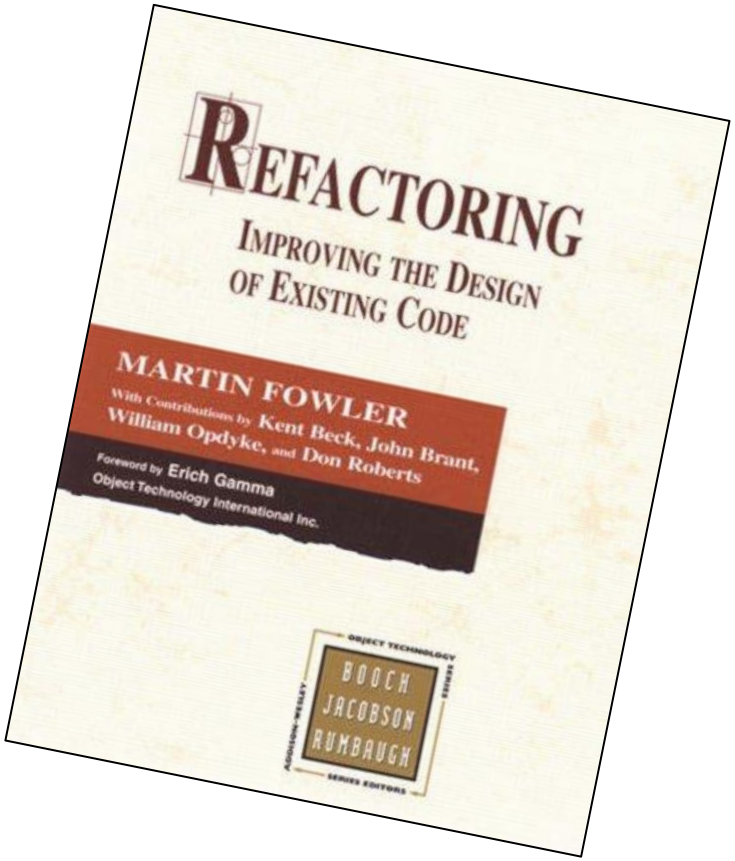
Checkstyle

FindBug

Bibliografía

Bibliografía “obligatoria”


PRENTICE HALL
Robert C. Martin Series
Clean Code
A Handbook of Agile Software Craftsmanship
Robert C. Martin
Foreword by James O. Coplien


REFACTORING
IMPROVING THE DESIGN OF EXISTING CODE
MARTIN FOWLER
With Contributions by Kent Beck, John Brant, William Opdyke, and Don Roberts
Foreword by Erich Gamma
Object Technology International Inc.
BOOCH JACOBSON RUMBAUGH
OBJECT TECHNOLOGY SERIES
SERIES EDITORS
ADDISON-WESLEY

Bibliografía adicional

The Addison-Wesley Signature Series
Refactoring to Patterns
Joshua Kerievsky
Implementation Patterns
Kent Beck
Object-Oriented Metrics in Practice
Michele Lanza
Radu Marinescu
Springer
Smalltalk Best Practice Patterns
Kent Beck
Code Complete
A practical handbook of software construction
Steve McConnell

ARGENTINA
Clay 2954
Capital Federal (C1426DLD)
tel: 54+11+5299 5400

Calle 48 Nº 1165 – 5º Piso B
La Plata,
Buenos Aires (CP 1900)

Belgrano 133 – 2º Piso
Bahía Blanca,
Buenos Aires (B8000IJC)

San Martín 902 – 1º Piso - Of. 6
Paraná,
Entre Ríos (E3100AAT)

BRASIL
Cardoso de Melo 1470 – 8, Vila Olimpia
San Pablo (04548004)
tel: 55+11+3045 2193

URUGUAY
Roque Graseras 857
Montevideo (11300)
tel: 598+2+7117879

USA
12105 Sundance Ct.
Reston (20194)
tel:+703 842 9455

www.hexacta.com

hexacta