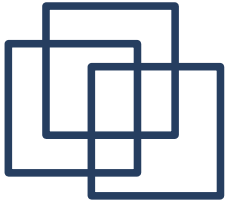


---

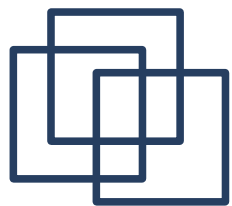
If debugging is the process of removing bugs,  
then programming must be the process of  
putting them in.

Edsger W. Dijkstra

---



# Ejemplo Motivador: Privacidad de Mensajes (CesarCipher)

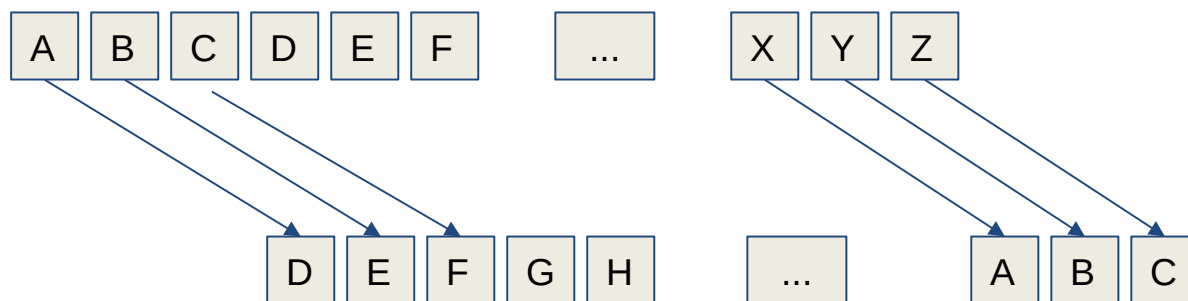


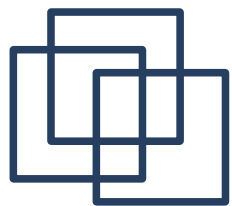
# Cifrado de Texto

---

Un texto cifrado es el resultado de aplicar un algoritmo que ha ofuscado un texto original (plain text)

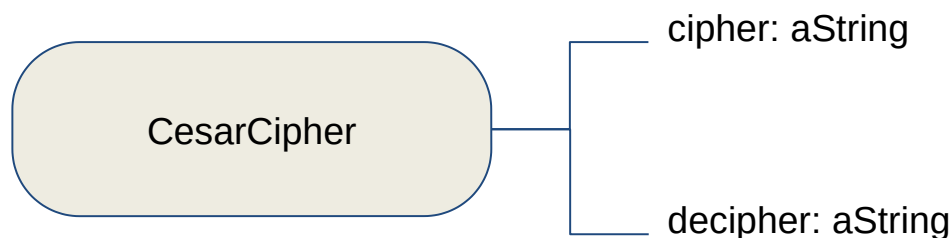
El cifrado de César es un cifrado de sustitución en el que una letra en el texto original es reemplazada por otra letra que se encuentra un número fijo de posiciones más adelante en el alfabeto.





# Cifrado Texto: Objetos

---



Considerando que existe un objeto que implementa el cifrado de César (`CesarCipher`) y que el objeto entiende dos mensajes:

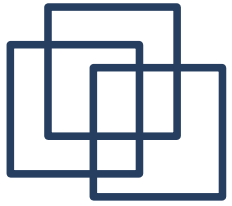
- `cipher:`
- `decipher:`

Cómo deberíamos testear que la implementación del cifrado de César es correcta?

---

A satellite is shown in orbit against the black background of space. The satellite has a central cylindrical body with various instruments and antennas. Two large, rectangular solar panels are extended from the sides, and two smaller ones are visible further back. The Earth's blue and white cloud-covered surface is visible in the lower-left portion of the frame.

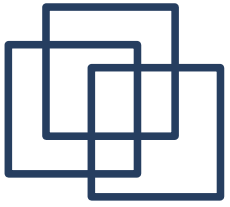
Testing de Unidad con SUnit



# Generación de Bugs

---

- Comunicación imperfecta
  - Optimismo mágico
    - Cambios menores, grandes errores
-



# Comunicación Imperfecta

---

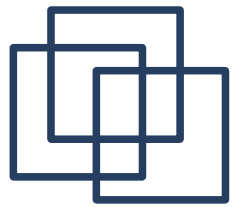
- Humano-Humano

*“Se requiere implementar la Ec. de Heron”*

$$(S (S - x) (S - y) (S - z) )^{1/2} ;$$

$$S = \frac{x + y + z}{2}$$

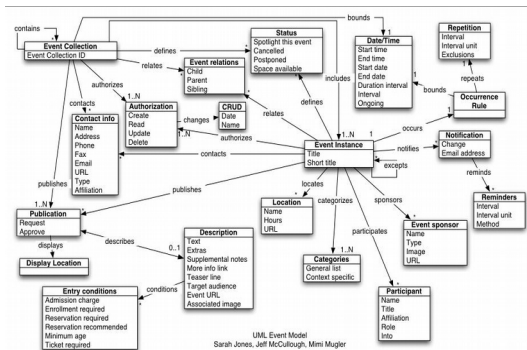




# Comunicación Imperfecta

## ● Modelo-Humano-Programa

Diseño avanzado con metodología X y programadores sin entrenamiento en metodología X...



```
MessageNotUnderstood: ByteString>>asFraction
ByteString(Object)>>doesNotUnderstand: #asFraction
UndefinedObject>>Dolt
Compiler>>evaluate:in:to: notifying:ifFail:logged:
[] in TextMorphForShoutEditor(ParagraphEditor)>>evaluateSelection
BlockClosure>>on:do:
TextMorphForShoutEditor(ParagraphEditor)>>evaluateSelection
[] in PluggableShoutMorph(PluggableTextMorph)>>dolt

stephane.ducasse 12/22/2008 13:53 · error handling · 14 implementors · in no change set ·

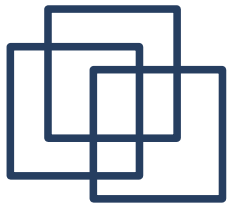
Proceed Restart Into Over Through Full Stack Run to Here Where

[Object new blah + 1]
on: MessageNotUnderstood
do: [:e | e resume: 1] does not loop indefinitely"

| exception resumeValue |
(exception := MessageNotUnderstood new)
message: aMessage;
receiver: self.

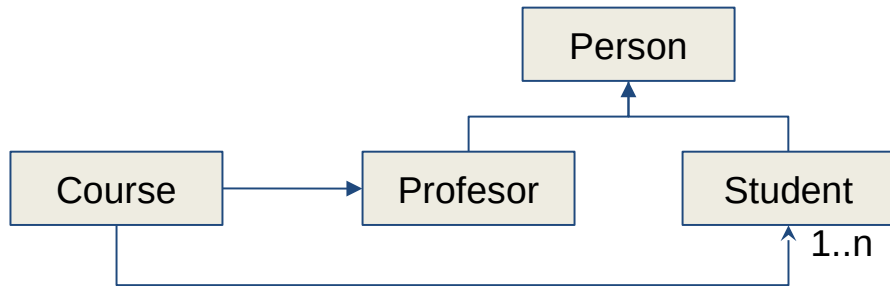
self
all inst vars
1
2

thisContext
stack top
all temp vars
aMessage
```

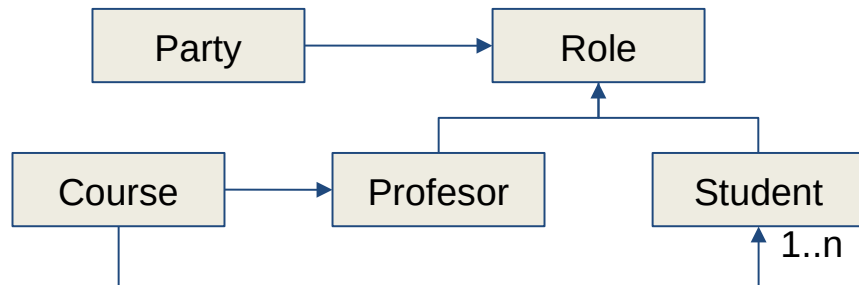
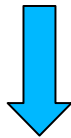


# Optimismo Mágico

“Esto funcionaba bien y es casi lo que necesitamos”



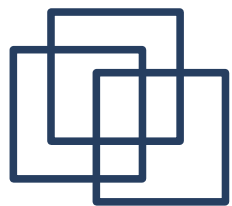
Alguien es Estudiante o Profesor



Alguien puede ser Estudiante y Profesor...  
pero tambien puede ser su propio Profesor



Congelamiento de O'rings



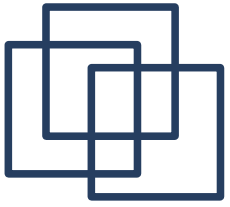
# Testing sistemático

---

- Determinar que hace un artefacto
- Determinar que no hace un artefacto
- Mantener funcionalidad minima
- Tener una medida de avance
- Tener una medida de éxito
- Estimaciones más realistas



© Mark Parisi, Permission required for use.



# El artefacto debe:

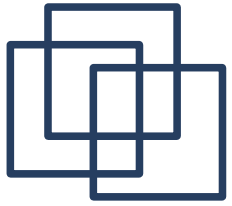
---

Hacer lo que se espera



No hacer lo que no se espera

---



# Paradoja del Testing

---

- Escribir casos de testing es deseable
- Escribir casos de testing costoso y aburrido(?)
- Testear todos los métodos no es práctico

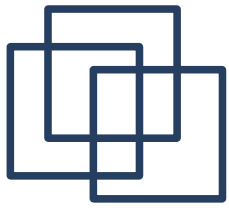
– Ej:

```
name  
^name
```

```
name: aString  
name := aString
```

**Objetivo: min. los casos y max. 'cobertura'**

---



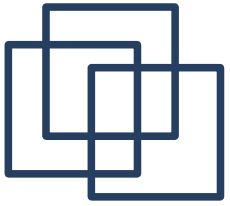
# Particiones Equivalentes

---

- Definir conjuntos de datos
    - si un elemento pasa el test, otros del mismo conjunto pasarán

*{CesarCipher}*  
*{CesarCipher de caracteres que no producen “loopback”}*  
*{CesarCipher de caracteres que producen “loopback”}*
  - Asegurarse que el artefacto no hace lo que no debe

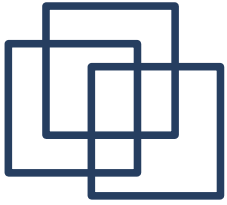
*CesarCipher no agrega o elimina caracteres*
-



# Valores de Borde

---

- La mayoría de los errores ocurren en los bordes o límites entre conjuntos
    - Cifrado de caracteres que producen loopback
    - Cifrado con alfabeto nulo
    - Cifrado con cadena de texto vacía
  - Cuál es el comportamiento esperado si el carácter no es parte del alfabeto?
  - División por cero es un borde “computacional”.
-



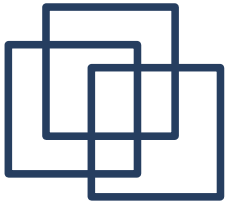
---

# Sunit

## -Framework for Unit Testing-

---

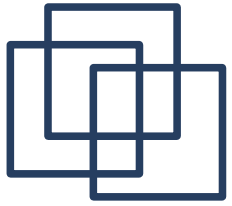




# SUnit

---

- Framework de testing para desarrollar 'test cases' en Smalltalk
  - Originalmente desarrollado por Kent Beck.
  - Existen implementaciones para muchos lenguajes.
    - JUnit, CUnit, PyUnit, etc
  - Existen extensiones testear ventanas y tambien web applications.
-



# Sunit's Customization

---

Crear subclase de `TestCase`

Realizar toda inicialización necesaria en `setUp`

Realizar toda limpieza necesaria en `tearDown`

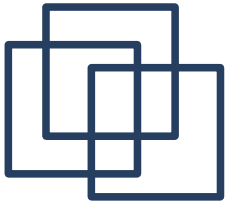
Para la funcionalidad Y en la clase X, tener un método `#testY` en la clase `TestX`.

Usar:

a) `self.assert t : (expresión booleans).`

b) `self.deny : (expresión booleans).`

---



# Crear una clase

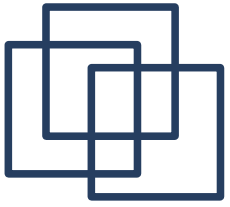
---

```
TestCase subclass: #CesarCipherTest  
  instanceVariableNames: 'cipher'  
  classVariableNames: ''  
  category: 'CipherTest'
```



Mars Global Surveyor

---



# setUp y tearDown

---

## setUp

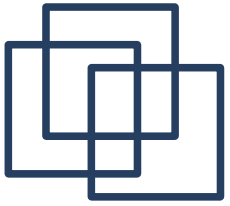
```
cipher := self cesarCipher
```

---

## tearDown

```
cipher := nil
```



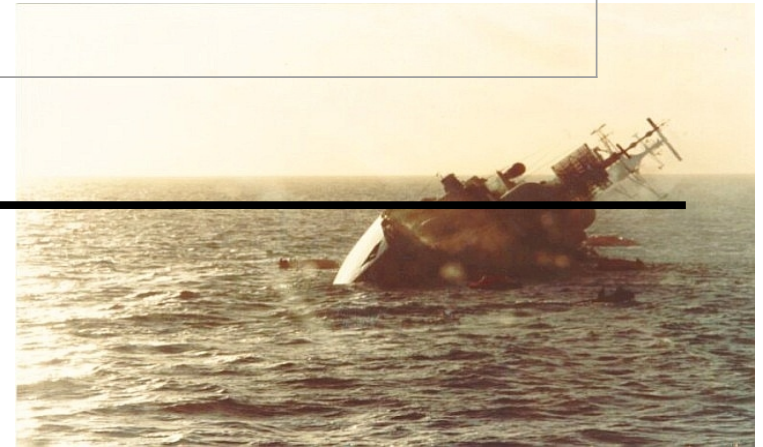


# Casos de testing

---

## **testCipherCharacters**

self assert: (cipher cipher: 'X') equals: 'A'.  
self assert: (cipher cipher: 'A') equals: 'D'.  
self assert: (cipher cipher: 'W') equals: 'Z'

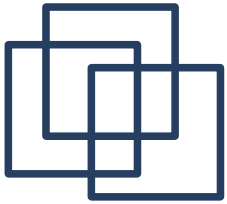


---

## **testCipherDecipher**

self assert: (cipher decipher: (cipher cipher: 'WIKIPEDIA'))  
  
equals: 'WIKIPEDIA'

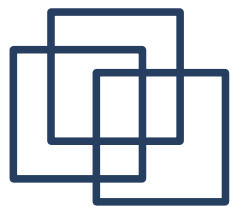
---



# Resumen

---

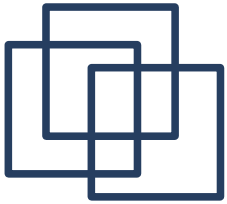
- Test Unidad implementa un mecanismo para probar la implementacion de métodos de un objetos
  - Cada test case debería implementar 1 aspecto de la funcionalidad
  - Todos los test cases conforman una “libreria” de pruebas con valor para:
    - dar por terminada una tarea => da una medida tangible de avance
    - para hacer test de regresión
    - ayudar en la integración de versiones (branches) en el repositorio de codigo
  - **Particiones Equivalentes es complementario de “Valores de Borde”**
-



## En la practica de OO1...

---

- Cesar Cipher es un tipo particular de Shift Cipher
  - Shift Cipher es un cifrador por substitucion que generaliza a CesarCipher dado que se puede parametrizar:
    - Alfabeto
    - Desplazamiento de Substitución
  - Alternativas de programar:
    - CesarCipher
    - ShiftCipher parametrizable a un CesarCipher
    - El test case es aplicable para ambas implementaciones
-



# Comm Check...: The Final Flight of Shuttle Columbia

Michael Cabbage, William Harwood, 2008

---





Charla: 18 Octubre Aula 1-4 18hs



# Lenguajes, Parsers y Analisis de Programas

## Introducción, Actualidad y Tendencias