

# #iloveruby

*“Una historia de amor”*

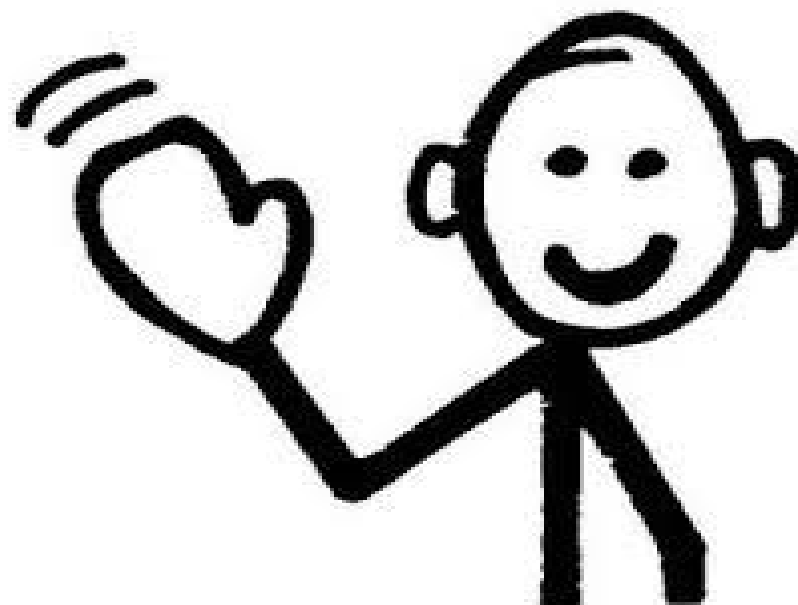
*para todos los públicos...*

CC-BY-SA-NC



# \$ whoami

```
me = {  
  :name => "David Vargas Ruiz",  
  :studies => "Ing. Informático ULPGC",  
  :job => "Profesor FP Informática",  
  :twitter => @dvarrui,  
  :github => dvarrui,  
  :email => dvarrui@protonmail.com  
}
```



# \$ history

Hablaremos de:

- ¿Por qué?
- ¿Cómo empezó todo?
- Características
- Ejemplos

No esperas al final para  
Preguntar. ;-)



# ¿Por qué?

- Para entender mejor Rails.
- Dar a conocer a Ruby.
- Crear comunidad.
- Compartir.
- Debatir y aprender.

*“El amor es lo que nos mueve...”*



# \$ whereis ruby

- 2000-2005 Java.
- JBoss, EJB, NetBeans.
- 2005 Charla Rails ULL.



*“Usa tu lenguaje favorito para todo.*

*Cuando te empiece a molestar...*

*Es que ha llegado el momento de cambiar”*

## \$ find . / -name “pareja”

*Los lenguajes de programación son como las parejas. Debes buscar hasta encontrar la adecuada. Con las características que te hacen feliz.*

*Cuando la encuentres lo sabrás porque lo querrás hacer todo con ella (GUI, CLI, Web, scripting, etc.)*

*Hay un lenguaje ideal para cada uno. Porque todos somos diferentes.*

# \$ uname -a



Mis requisitos:

- Licencia Libre
- Multiplataforma
- Multiparadigma
- POO
- Sencillo
- ... *y con “magia”*

# zypper in





\$ su



# Estructura secuencial (I)

```
#Ejemplo 1  
#Sumar enteros  
a = 3  
b = 4  
puts a + b  
# 7
```

```
#Ejemplo 2  
#Unir Strings  
a = "Hello"  
b = "World!"  
puts a + b  
# "HelloWorld"
```

*Rule1: Declaración de variables dinámica.*

## Estructura secuencial (II)

```
#Sumar enteros con Strings
```

```
a = "1"
```

```
b = 0
```

```
puts a + b
```

```
# ¿10 o "10"?
```

## Estructura secuencial (II)

```
# Sumar enteros con Strings ???
```

```
a = "1"
```

```
b = 0
```

```
puts a + b
```

```
# ¿10 o "10"?
```

```
# Error: No implicit conversion of Integer into String
```

*Rule2: Tipado fuerte.*

# Características

- Lenguaje interpretado
- Declaración dinámica de variables
- Tipado fuerte
- Fácil de leer
- Buena documentación (rdoc)
- Muchas bibliotecas (rubygems)
- Muy expresivo. No necesita IDE...

*¡Lo sé! Aquí vamos a discutir*

# Estructura condicional

#Ejemplo 1

```
edad = gets
if (edad > "18")
  puts "Eres mayor"
end
```

#Ejemplo 2

```
edad = gets.to_i
if edad > 18
  puts "Eres mayor"
end
```

#Ejemplo 3

```
edad = gets.to_i
puts "Eres mayor" if (edad > 18)
```

# Estructura iterativa: los bucles (I)

#Ejemplo 1

```
i = 0
```

```
while (i<10)
```

```
  puts i
```

```
  i++
```

```
end
```

# <i> es un objeto

# de tipo Integer

#Ejemplo 2

```
for i in 0..9
```

```
  puts i
```

```
end
```

# <0..9> es un objeto

# de tipo Range

## Estructura iterativa: los bucles (II)

#Ejemplo 1

```
10.times do |i|  
  puts i  
end
```

# <times> es un  
# método del objeto <10>  
# do ... end es un bloque.  
# Los bloques son objetos

#Ejemplo 2

```
(0..9).each { |i| puts i }
```

# <each> es un método  
# del Range <0..9>  
# { ... } es un bloque  
# también.



# Estrategias para programar



- Estructuras de control (Secuencial, condicional, repetitiva)
- Programacion Estructurada: Subrutinas y 3 estructuras básicas.
- Programación modular: dividir en subprogramas.
- POO (Programación Orientada a Objetos)
- Diseño de Patrones

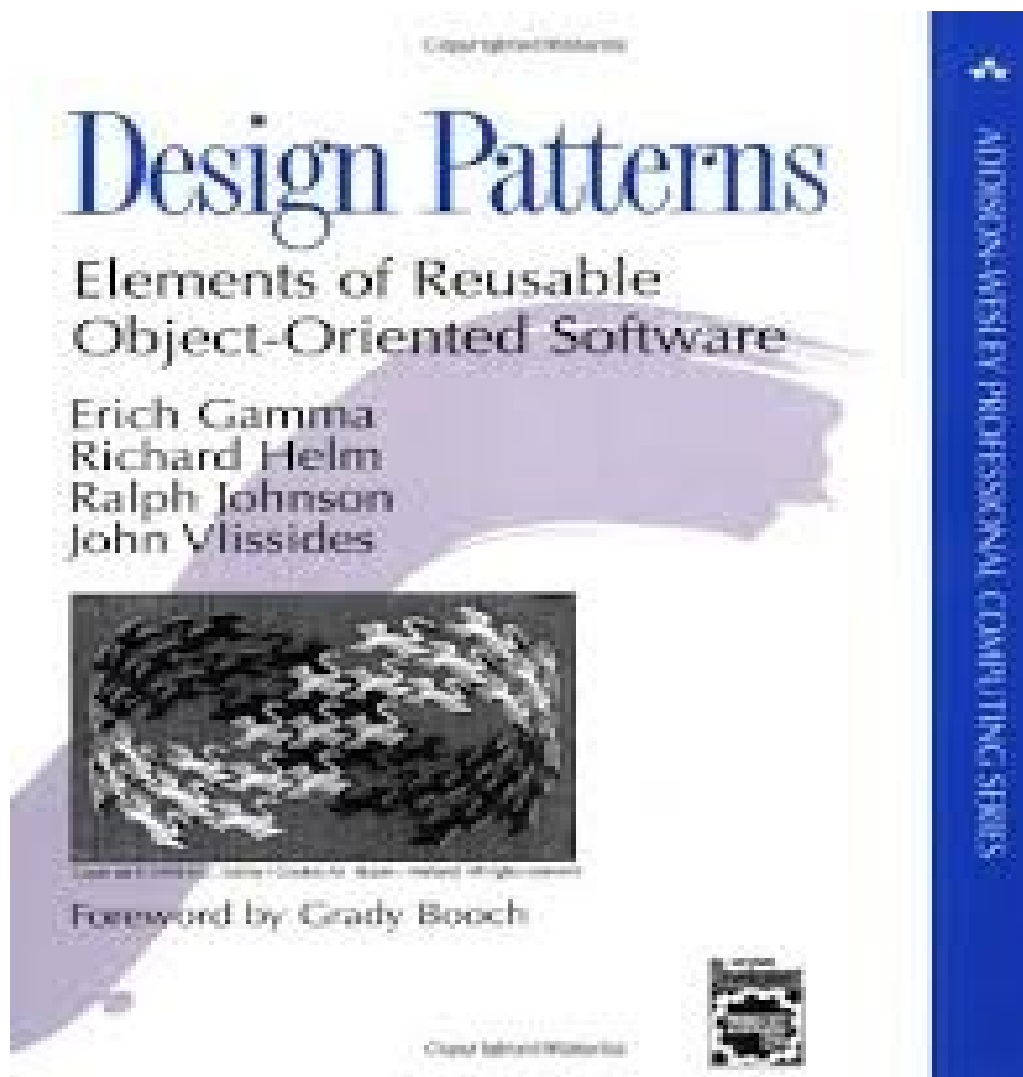
# POO: Clases, instancias, atributos

```
#Ejemplo 1
class Persona
  def initialize(name)
    @name = name
  end
  def saludar
    puts "Hola, me llamo" + @name
  end
end
```

## Ver ejemplo de P00



# Libro “Patter design”, (The Gang of four)



- MVC
- ITERADOR
- Singleton
- Decorador
- Builder
- Observer
- Template
- Wrapper
- Etc.

## Estructura iterativa: los bucles (III)

#Ejemplo 1

```
a = [3, 14, 27]
```

```
a.each do |i|
```

```
  puts i
```

```
end
```

# <each> es un

#método del objeto a

# a es un objeto Array

#Ejemplo 2

```
a = Array.new
```

```
a << 3
```

```
a << 14
```

```
a << 24
```

```
puts a
```

# a es [3,14,27]

# a es un objeto Array

# Con Ruby salgo a todos lados...

- GUI, CLI, Web (Cliente y Server), Scripting, Etc



# Hagamos algo ahora...



**¡Y vivieron felices y comieron perdices!**





# \$ whoami

```
me = {  
  :name => "David Vargas Ruiz",  
  :studies => "Ing. Informático ULPGC",  
  :job => "Profesor FP Informática",  
  :twitter => @dvarrui,  
  :github => dvarrui,  
  :email => dvarrui@protonmail.com  
}
```

