

Metodología para la Enseñanza Aprendizaje de la Lógica de la Programación Orientada a Objetos


Leobardo López Román

llopez@industrial.uson.mx

Universidad de Sonora

Hermosillo, Sonora, México

A G E N D A

- 1. INTRODUCCIÓN**
 - 2. EVOLUCIÓN DE LA PROGRAMACIÓN**
 - 3. PROBLEMÁTICA DE LA ENSEÑANZA-APRENDIZAJE DE LA PROGRAMACIÓN ORIENTADA A OBJETOS**
 - 4. METODOLOGÍA DE LA PROGRAMACIÓN ORIENTADA A OBJETOS**
 - 5. CONCLUSIONES**
- 

1. INTRODUCCIÓN

Plantear

que la Enseñanza-Aprendizaje de la
Programación Orientada a Objetos
usando Java
no se esta haciendo en forma apropiada

- - - - -

Mostrar

una metodología
que ayude a mejorar dicha situación

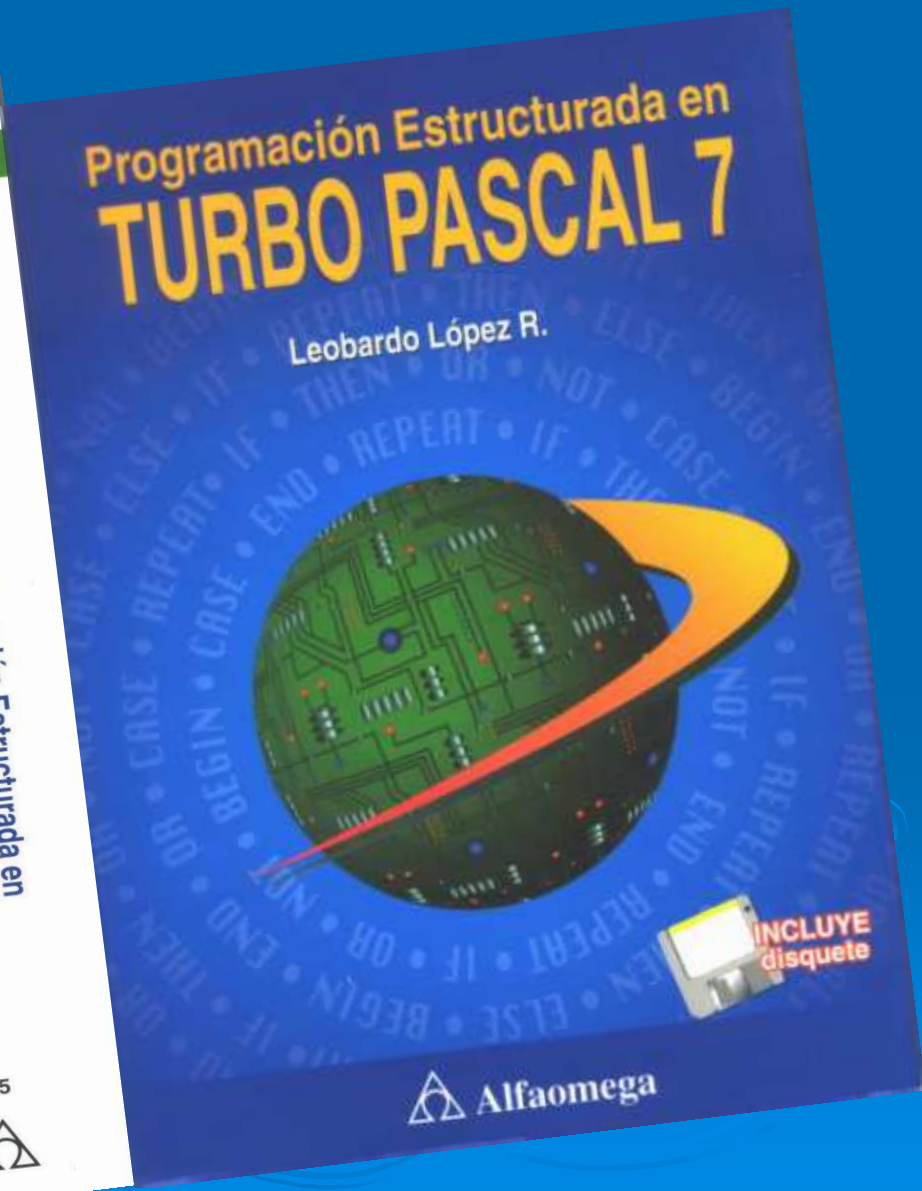
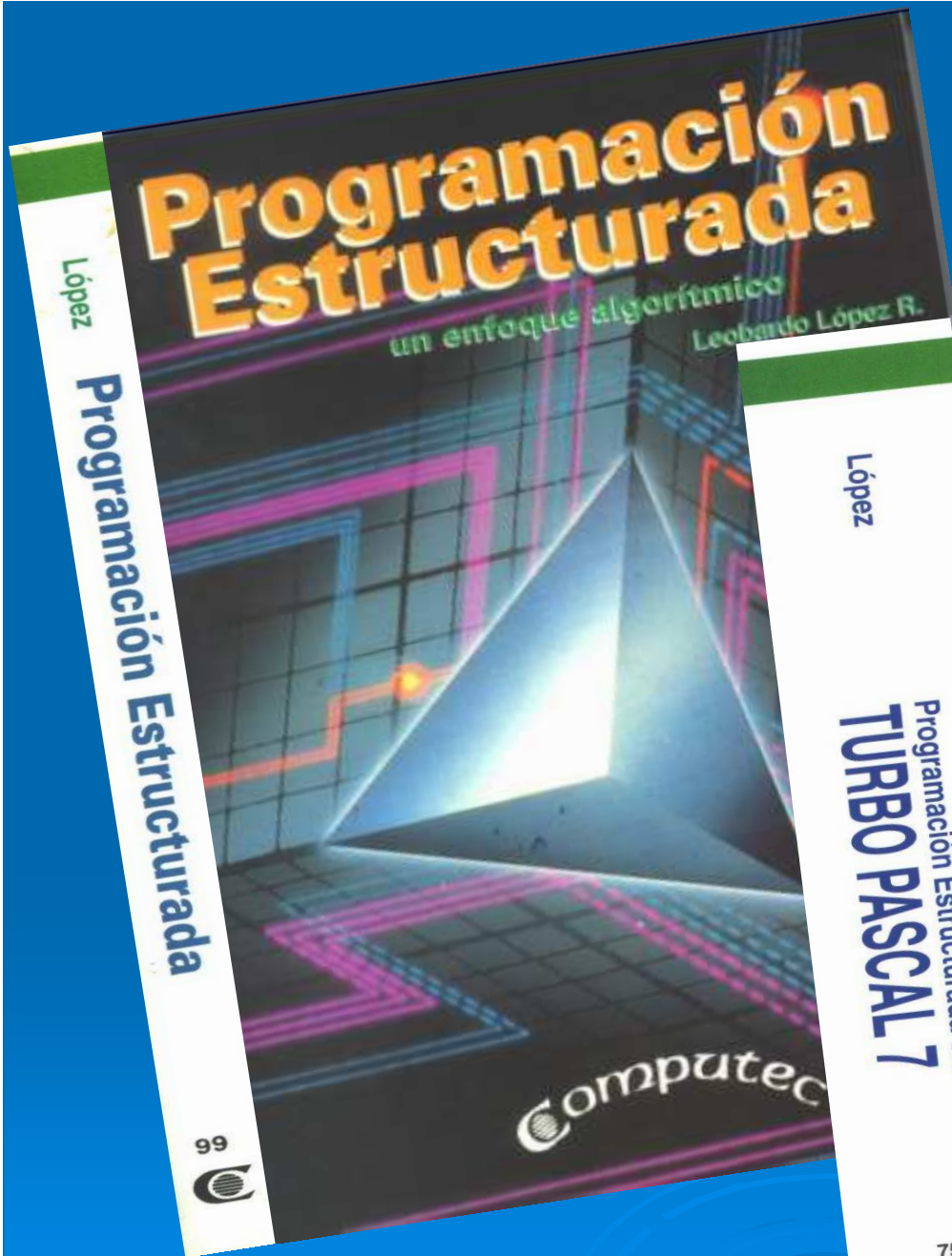
2. EVOLUCIÓN DE LA PROGRAMACIÓN

PROGRAMACIÓN ORIENTADA A OBJETOS

PROGRAMACIÓN ESTRUCTURADA

PROGRAMACIÓN
TRADICIONAL

The diagram illustrates the evolution of programming through three nested rectangular boxes. The outermost box is the largest and contains the text 'PROGRAMACIÓN ORIENTADA A OBJETOS'. Inside it is a smaller box containing 'PROGRAMACIÓN ESTRUCTURADA'. The innermost box is the smallest and contains 'PROGRAMACIÓN TRADICIONAL'. The boxes are centered and nested, showing a progression from the most traditional to the most modern programming paradigm. The background of the slide is blue with a subtle pattern of concentric circles at the bottom.



López

Programación Estructurada

UN ENFOQUE ALGORÍTMICO

2ª Edición

Leobardo López Román

Programación Estructurada
UN ENFOQUE ALGORÍTMICO

856



Alfaomega

López

Programación estructurada en lenguaje C

Leobardo López Román

Incluye CD



Programación estructurada en lenguaje C

1062



Alfaomega

En la actualidad

Java

Que es un Lenguaje Orientado a Objetos

**Se esta utilizando cada vez más
como primer lenguaje**



3. PROBLEMÁTICA DE LA ENSEÑANZA-APRENDIZAJE DE LA PROGRAMACIÓN ORIENTADA A OBJETOS

- - - - -

Se enseña Java directamente

Sin una formación previa que desarrolle la lógica

Aprenden Java a prueba y error

Se están formando codificadores; no programadores

- - - - -

Los principiantes si entienden los conceptos POO en abstracto; pero, a la hora de programar no pueden

Los estudiantes cursan Java, pero sienten que no aprendieron

¿ Cuál es la causa ?

**Que en la bibliografía
esta ausente la
Metodología de la
Programación Orientada a Objetos
enfocada
a niveles básicos de aprendizaje**

**Libros
Análisis, Diseño,
Modelado, UML, . .
OO**

**Libros
“Manuales”
de
Java**

López

METODOLOGÍA DE LA Programación Orientada a Objetos

Leobardo López Román

Metodología de la Programación
Orientada a Objetos



CONTIENE
CD-ROM
con ejercicios
y programas
en Java

1173

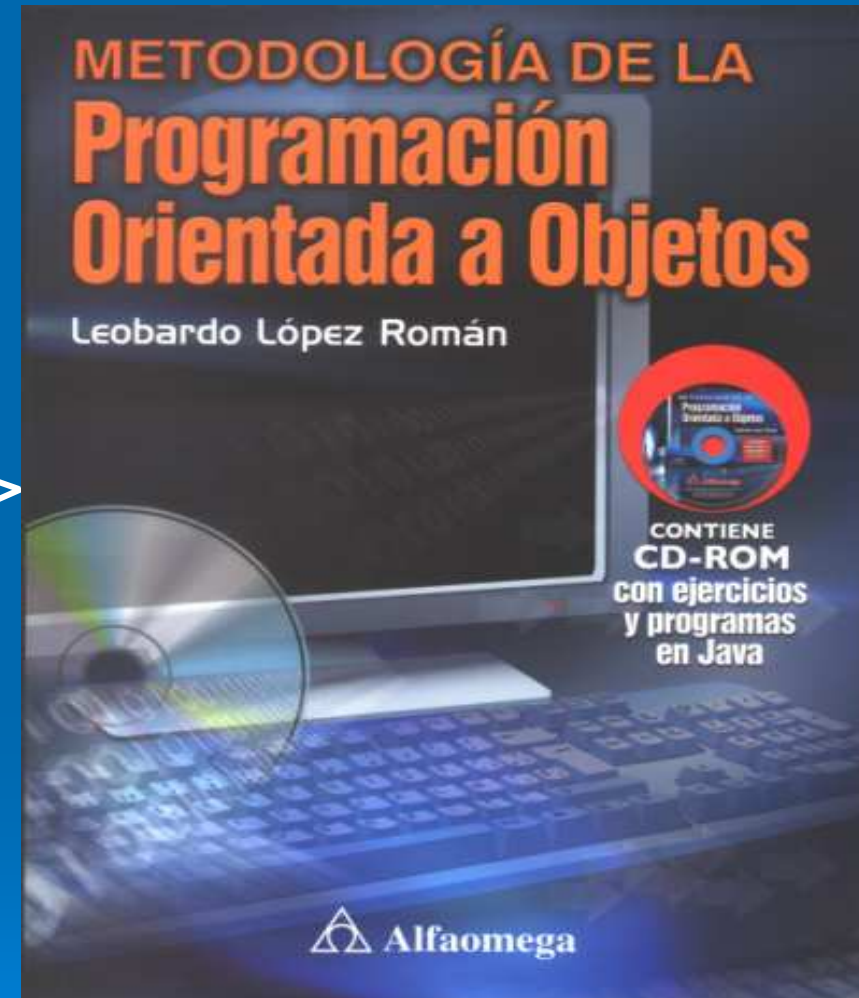


Alfaomega

LA METODOLOGIA PROPUESTA

Programación
Orientada a
Objetos

Programación
Estructurada



4. METODOLOGÍA DE LA PROGRAMACIÓN ORIENTADA A OBJETOS

4.1 Primera parte (Capítulos 1 - 9)

Se explica como desarrollar la lógica básica usando pseudocódigo:

- Estructuras de datos: Tipos de datos, variables, constantes
- Operaciones primitivas elementales: Entrada, proceso y salida
- Estructuras lógicas de control
 - La secuenciación
 - La selección IF THEN, IF THEN ELSE y SWITCH
 - La repetición DO...WHILE, FOR y WHILE
- Arreglos
- Métodos
 - Que no regresan valor
 - Que regresan valor
 - Parámetros por valor y por referencia

Ejemplos:

Algoritmo CALCULA SUELDO DE UN EMPLEADO

Clase Empleado1

1. Método principal

a. Declaraciones

Variables

nombreEmp: Cadena

horasTrab: Entero

cuotaHora, sueldo: Real

b. Solicitar Nombre del empleado, número de horas trabajadas
y cuota por hora

c. Leer nombreEmp, horasTrab, cuotaHora

d. Calcular sueldo = horasTrab * cuotaHora

e. Imprimir nombreEmp, sueldo

f. Fin Método principal

Fin Clase Empleado1

Fin

Algoritmo CALCULO SUELDO DOBLE

Clase Empleado2

1. Método principal

a. Declaraciones

Variables

nombreEmp: Cadena

horasTrab: Entero

cuotaHora, sueldo: Real

b. Solicitar Nombre del empleado, número de horas trabajadas y cuota por hora

c. Leer nombreEmp, horasTrab, cuotaHora

d. IF horasTrab <= 40 THEN

1. sueldo = horasTrab * cuotaHora

e. ELSE

1. sueldo = (40*cuotaHora) + ((horasTrab-40)*(cuotaHora*2))

f. ENDIF

g. Imprimir nombreEmp, sueldo

h. Fin Método principal

Fin Clase Empleado2

Fin

Algoritmo OBREROS

Clase Obreros1

1. Método principal

a. Declaraciones

Variables

nombreObr, obrMayor, obrMenor: Cadena
proDia, totProd, toTotProd, totObreros,
mayorProd, menorProd: Entero
otro, desea: Carácter

b. Imprimir encabezado

c. totObreros = 0

toTotProd = 0

mayorProd = 0

menorProd = 10000

d. DO

1. Solicitar Nombre

2. Leer nombreObr

3. totProd = 0

4. DO

a. Solicitar Producción del día

b. Leer proDia

c. totProd = totProd + proDia

d. Preguntar "¿Desea procesar otro día (S/N)?"

e. Leer otro

5. WHILE otro = "S"

6. Imprimir nombreObr, totProd

```
7. IF totProd > mayorProd THEN
    a. mayorProd = totProd
    b. obrMayor = nombreObr
8. ENDIF
9. IF totProd < menorProd THEN
    a. menorProd = totProd
    b. obrMenor = nombreObr
10. ENDIF
11. totObreros = totObreros + 1
    toTotProd = toTotProd + totProd
12. Preguntar "¿Desea procesar otro obrero (S/N)?"
13. Leer desea
e. WHILE desea = "S"
f. Imprimir totObreros, toTotProd, obrMayor, mayorProd,
    obrMenor, menorProd
g. Fin Método principal
Fin Clase Obreros1
Fin
```


Algoritmo MEDIA CON METODOS

Clase MediaConMetodos

1. Método principal

a. Declaraciones

Variables

vector: Arreglo[10] Real

promedio: Real

b. Llamar leerVector(vector)

c. promedio = calcularMedia(vector)

d. Llamar imprimirVector(vector)

e. Imprimir promedio

f. Fin Método principal

2. Método leerVector(Ref vec: Arreglo[10] Real)

a. Declaraciones

Variables

n: Entero

b. FOR n=0; n<=9; n++

1. Solicitar elemento vec[n]

2. Leer vec[n]

c. ENDFOR

d. Fin Método leerVector

3. Método calcularMedia(Ref v: Arreglo[10] Real): Real

a. Declaraciones

Variables

sumatoria, prom: Real

i: Entero

b. sumatoria = 0

c. FOR i=0; i<=9; i++

1. sumatoria = sumatoria + v[i]

d. ENDFOR

e. prom = sumatoria / i

f. return prom

g. Fin Método calcularMedia

4. Método imprimirVector(Ref vect: Arreglo[10] Real)

a. Declaraciones

Variables

x: Entero

b. FOR x=0; x<=9; x++

1. Imprimir vect[x]

c. ENDFOR

d. Fin Método imprimirVector

Fin Clase MediaConMetodos

Fin

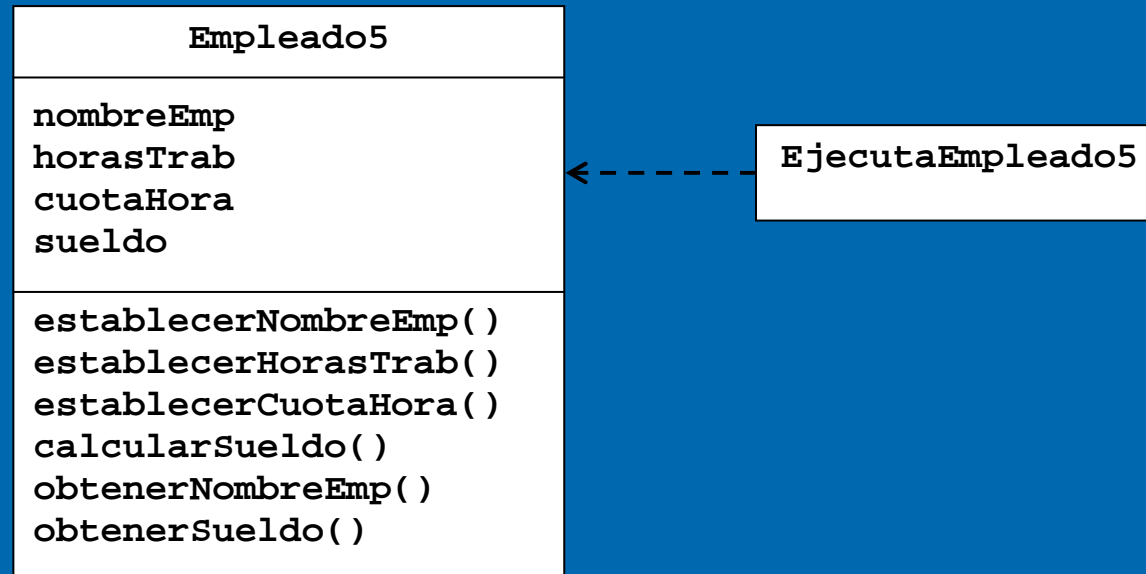
4. METODOLOGÍA DE LA PROGRAMACIÓN ORIENTADA A OBJETOS

4.2 Segunda parte (Capítulos 10 - 16)

Se explica como desarrollar la lógica orientada a objetos usando pseudocódigo, integrando:

- Los conceptos de la programación orientada a objetos
 - Objetos, clases, encapsulación
 - Herencia y polimorfismo
- El diagrama de clases de UML (Unified Modeling Language)
- La arquitectura modelo-vista-controlador
- Algunos conceptos introducidos por el lenguaje Java
- Las bases lógicas de la programación desarrolladas en la primera parte

Diagrama de clases



Algoritmo CALCULAR SUELDO DE UN EMPLEADO

Clase Empleado5

1. Declaraciones

Datos

nombreEmp: Cadena

horasTrab: Entero

cuotaHora: Real

sueldo: Real

2. Método establecerNombreEmp(nom: Cadena)

a. nombreEmp = nom

b. Fin Método establecerNombreEmp

3. Método establecerHorasTrab(horasTr: Entero)

a. horasTrab = horasTr

b. Fin Método establecerHorasTrab

4. Método establecerCuotaHora(cuotaHr: Real)

a. cuotaHora = cuotaHr

b. Fin Método establecerCuotaHora

5. Método calcularSueldo()

a. sueldo = horasTrab * cuotaHora

b. Fin Método calcularSueldo

6. Método obtenerNombreEmp(): Cadena

a. return nombreEmp

b. Fin Método obtenerNombreEmp

- 7. Método obtenerSueldo(): Real
- a. return sueldo
- b. Fin Método obtenerSueldo

Fin Clase Empleado5

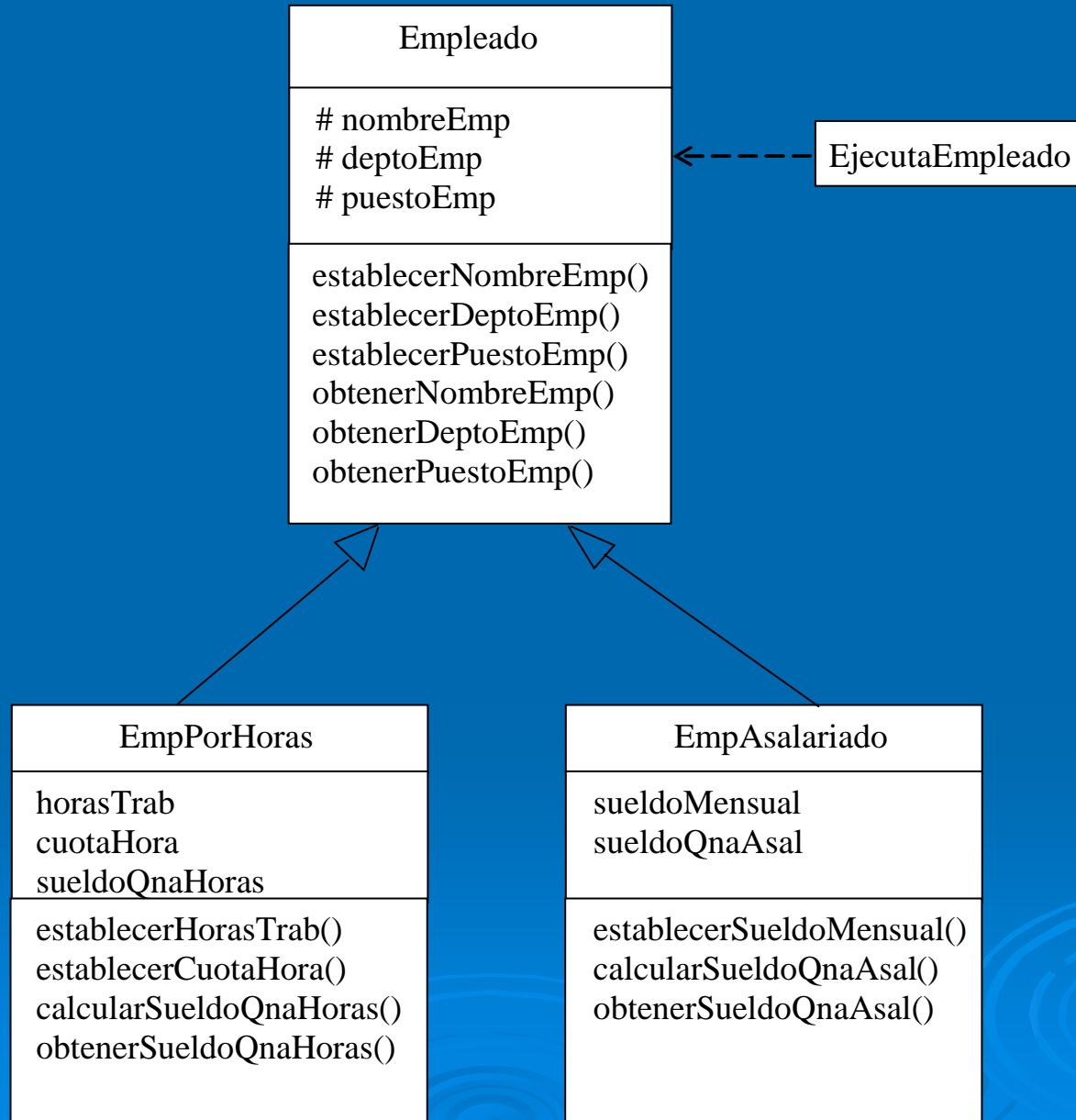
Clase EjecutaEmpleado5

- 1. Método principal
 - a. Declaraciones
 - Variables
 - nomEmp: Cadena
 - hrsTra: Entero
 - cuoHr: Real
 - b. Declarar, crear e iniciar objeto
 - c. Solicitar Nombre, número de horas trabajadas y cuota por hora
 - d. Leer nomEmp, hrsTra, cuoHr
 - e. Establecer objEmpleado.establecerNombreEmp(nomEmp)
objEmpleado.establecerHorasTrab(hrsTra)
objEmpleado.establecerCuotaHora(cuoHr)
 - f. Calcular objEmpleado.calcularSueldo()
 - g. Imprimir objEmpleado.obtenerNombreEmp()
objEmpleado.obtenerSueldo()
 - h. Fin Método principal

Fin Clase EjecutaEmpleado5

Fin

Diagrama de clases



Algoritmo CALCULAR SUELDO DE VARIOS EMPLEADOS

Clase Empleado

1. Declaraciones

Datos

- # nombreEmp: Cadena
- # deptoEmp: Cadena
- # puestoEmp: Cadena

2. Método establecerNombreEmp(nom: Cadena)

- a. nombreEmp = nom
- b. Fin Método establecerNombreEmp

3. Método establecerDeptoEmp(dep: Cadena)

- a. deptoEmp = dep
- b. Fin Método establecerDeptoEmp

4. Método establecerPuestoEmp(pue: Cadena)

- a. puestoEmp = pue
- b. Fin Método establecerPuestoEmp

5. Método obtenerNombreEmp(): Cadena

- a. return nombreEmp
- b. Fin Método obtenerNombreEmp

6. Método obtenerDeptoEmp(): Cadena

- a. return deptoEmp
- b. Fin Método obtenerDeptoEmp

7. Método obtenerPuestoEmp(): Cadena

- a. return puestoEmp
- b. Fin Método obtenerPuestoEmp

Fin Clase Empleado

Clase EmpPorHoras hereda de Empleado

1. Declaraciones

Datos

horasTrab: Entero
cuotaHora: Real
sueldoQnaHoras: Real

2. Método establecerHorasTrab(horasTr: Entero)

- a. horasTrab = horasTr
- b. Fin Método establecerHorasTrab

3. Método establecerCuotaHora(cuotaHr: Real)

- a. cuotaHora = cuotaHr
- b. Fin Método establecerCuotaHora

4. Método calcularSueldoQnaHoras()

- a. $\text{sueldoQnaHoras} = \text{horasTrab} * \text{cuotaHora}$
- b. Fin Método calcularSueldoQnaHoras

5. Método obtenerSueldoQnaHoras(): Real

- a. return sueldoQnaHoras
- b. Fin Método obtenerSueldoQnaHoras

Fin Clase EmpPorHoras

Clase EmpAsalariado hereda de Empleado

1. Declaraciones

Datos

sueldoMensual: Real

sueldoQnaAsal: Real

2. Método establecerSueldoMensual(sdo: Real)

a. sueldoMensual = sdo

b. Fin Método establecerSueldoMensual

3. Método calcularSueldoQnaAsal()

a. sueldoQnaAsal = sueldoMensual / 2

b. Fin Método calcularSueldoQnaAsal

4. Método obtenerSueldoQnaAsal(): Real

a. return sueldoQnaAsal

b. Fin Método obtenerSueldoQnaAsal

Fin Clase EmpAsalariado

Clase EjecutaEmpleado

1. Método principal

a. Declaraciones

Variables

nomEmp, depto, puesto: Cadena

hrsTra, tipoEmp: Entero

cuoHr, sdoMen: Real

desea: Carácter

b. DO

1. Imprimir Menu y solicitar tipo de empleado

Tipos de empleado

1. Empleado por horas

2. Empleado asalariado

Teclee tipo:

2. Leer tipoEmp

3. Solicitar Nombre, departamento y puesto

4. Leer nomEmp, depto, puesto

5. IF tipoEmp = 1 THEN

a. Crear objeto

EmpPorHoras objEmpleado = new EmpPorHoras()

b. Solicitar número de horas trabajadas y cuota por hora

c. Leer hrsTra, cuoHr

d. Establecer

objEmpleado.establecerNombreEmp(nomEmp)

objEmpleado.establecerDeptoEmp(depto)

objEmpleado.establecerPuestoEmp(puesto)

objEmpleado.establecerHorasTrab(hrsTra)

objEmpleado.establecerCuotaHora(cuoHr)

- e. Calcular `objEmpleado.calcularSueldoQnaHoras()`
- f. Imprimir `objEmpleado.obtenerNombreEmp()`
`objEmpleado.obtenerDeptoEmp()`
`objEmpleado.obtenerPuestoEmp()`
`objEmpleado.obtenerSueldoQnaHoras()`

6. ELSE

- a. Crear objeto
`EmpAsalariado objEmpleado = new EmpAsalariado()`
- b. Solicitar sueldo mensual
- c. Leer `sdoMen`
- d. Establecer
`objEmpleado.establecerNombreEmp(nomEmp)`
`objEmpleado.establecerDeptoEmp(depto)`
`objEmpleado.establecerPuestoEmp(puesto)`
`objEmpleado.establecerSueldoMensual(sdoMen)`
- e. Calcular `objEmpleado.calcularSueldoQnaAsal()`
- f. Imprimir `objEmpleado.obtenerNombreEmp()`
`objEmpleado.obtenerDeptoEmp()`
`objEmpleado.obtenerPuestoEmp()`
`objEmpleado.obtenerSueldoQnaAsal()`

7. ENDIF

8. Preguntar "¿Desea procesar otro empleado(S/N)?"

9. Leer desea

c. WHILE desea = "S"

d. Fin Método principal

Fin Clase EjecutaEmpleado

Fin

Cómo usar la Metodología de la Programación Orientada a Objetos

Semestre 1

Metodología de
la Programación
Orientada a
Objetos

Semestre 2

Lenguaje
Java

Semestre 3

Metodología de
la Programación
Orientada a
Objetos
Lenguaje Java


Semestre 4

Metodología de
la Programación
Estructurada

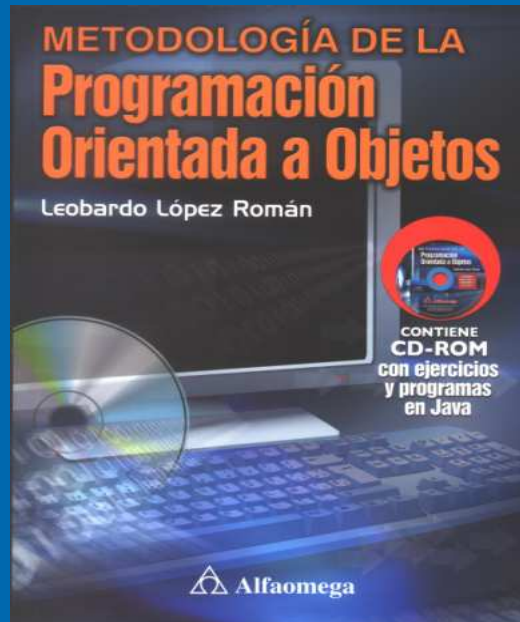
Lenguaje C

Lenguaje
Java (Cont.)

5. CONCLUSIONES (1)

1. La programación es lógica, independientemente de algún lenguaje.
 2. No se debe enseñar Java directamente.
 3. Primero, se debe desarrollar la lógica de la Programación Orientada a Objetos.
 4. El desarrollo de la lógica es un proceso mental formal; no es fácil, ni rápido, ni bonito.
- 

5. CONCLUSIONES (2)



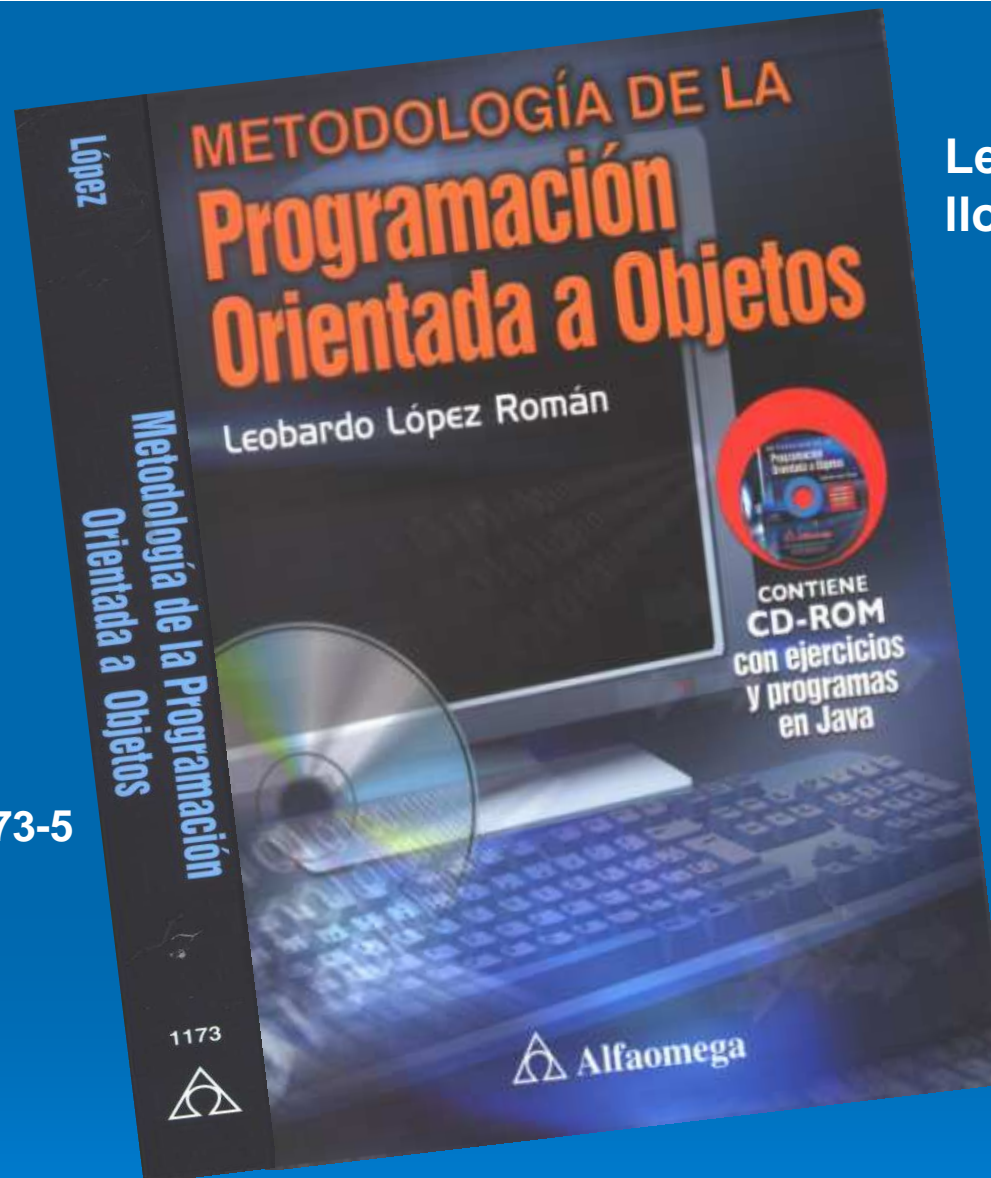
Ni muy abstracta

Libros
Análisis, Diseño,
Modelado, UML, . .
OO

Ni muy concreta

Libros
“Manuales”
de
Java

Sustentada en la lógica básica de la programación inmersa en la POO
Prepara a los estudiantes para aprender y comprender los lenguajes OO



Leobardo López Román
llopez@industrial.uson.mx

México, 2006
ISBN 970-15-1173-5

Distribuidor de Alfaomega (www.alfaomega.com.mx)
en España: Marcombo (www.marcombo.com)