



UNIVERSIDAD NACIONAL DE LA PLATA  
FACULTAD DE INFORMÁTICA

---

## METODOLOGÍAS DE PROGRAMACION

Año 2004

Carrera: **Analista de Computación P90**  
**Licenciatura en Informática P90**

Año: **2°**

Duración: **Anual**

Coordinador: **Dr. Gustavo Rossi**

Profesor: **Prof. Alicia Díaz**

**Prof. Roxana Giandini**

---

### Objetivos

El objetivo de Metodologías de Programación es introducir a los alumnos en técnicas modernas de la programación, en particular en aspectos de la programación visual, orientada a objetos y diseño orientado a objetos. Se enfatizará en la construcción de arquitecturas de software modulares, extensibles y reusables, conceptos claves para aplicaciones de gran porte. Se introducirá también al alumno en el uso de un lenguaje de modelado gráfico orientado a objetos (UML), que le permitirá construir diagramas especificando distintos aspectos de un sistema. Los trabajos prácticos se realizarán usando el lenguaje de modelado y diferentes lenguajes de implementación, tales como Smalltalk, Java, C++, Delphi, etc. que son los más apropiados de acuerdo a estos objetivos.

### Programa

1. La crisis del software. Problemas de las técnicas tradicionales (procedurales). Resolución de problemas complejos. El problema de la extensibilidad, el reuso y el mantenimiento.
2. Tipos Abstractos de Datos. Encapsulamiento. Information hiding.
3. Conceptos básicos: Objetos y Programa Orientado a Objetos. Comportamiento de un Objeto. Mensaje y Método.
4. Clasificación: Clases e Instancias. Instanciación.
5. Conceptos avanzados: Jerarquías de Clases. Relación *isA*. Generalización / Especialización. Herencia, Herencia Simple. Clases Abstractas.
6. Hacia mayor genericidad de código: polimorfismo, binding dinámico.
7. Diseño de objetos Complejos. Relaciones entre Objetos. Relación de *conocimiento*. Relación *isPartOf*.
8. Estructuras de datos como Objetos. Objetos contenedores.



9. Lenguajes orientados a objetos: variantes. El lenguaje Smalltalk. Tipos de Mensajes. Biblioteca de clases, jerarquías pre-definidas. Clases *Integer*, *Fraction*, *Point*, *Date*. Variables de instancia. PseudoVariables: *self* y *super*. Método *new*.
10. Clases *Boolean*, *False* y *True*. Métodos: *or:*, *and:* *not*.
11. Definición de bloques de código. Clase *Context*. Métodos: *value* y *value:*.
12. Estructuras de Control: Métodos *ifTrue:*, *ifFalse:*, *ifTrue: ifFalse:*, *whileTrue:*, *whileFalse:*.
13. Colecciones de Objetos. Clase *Collection* y sus subclases *Array*, *OrderedCollection*, *Set*, *Dictionary* y *SortedCollection*. Protocolo estándar.
14. Iteradores: *to: do:*, *to: by: do:*, *timesRepeat:*. El iterador *do:* de *Collection*. Otros iteradores de *Collection*: *select:*, *detect:*, *reject:*, *collect:*, *inject: to:*.
15. Lenguajes de modelado orientados a objetos: historia y variantes. El lenguaje de Modelado Unificado (*Unified Modeling Language*). Diagramas del UML. Introducción.
16. Diagramas de Estructura Estática: Diagramas de Paquetes y Diagramas de Clases
17. Diagramas Dinámicos ó de Comportamiento. Introducción
18. Diagramas de Interacción: Diagramas de Secuencia y Diagramas de Colaboración
19. Diagramas de Estado y de Actividades
20. Diagramas de Casos de Uso
21. Introducción al Proceso de Desarrollo (*Unified Process*). Ejemplos
22. Aplicación de técnicas orientadas a objetos para la construcción de aplicaciones con interfaces visuales. Mecanismos de Dependencias. Diseño de Interfaces: Model - View - Controller. Métodos *changed:* y *update:*.
23. Recursión: Definición de problemas recursivos. Conceptos básicos Ejemplos.
24. Diseño de soluciones recursivas usando un modelo Orientado a Objetos. Uso del *self*. Colaboración entre objetos que pertenecen a una misma jerarquía de clases.
25. El ciclo de desarrollo de aplicaciones usando objetos. Conceptos básicos de análisis, diseño y mantenimiento.
26. Reuso de componentes en el ambiente Smalltalk.
27. Diseño orientado a objetos. El enfoque basado en responsabilidades.
28. Optimización del diseño. Resolución de problemas complejos. Introducción a los patrones: *abstract factory*, *strategy*, *state*.

### **Bibliografía**

#### **Programación Orientada a Objetos**

1. B. Liskov, J. Guttag, *Abstraction and Specification in Program Development*, MIT Press, Mc Graw Hill, 1987. ISBN: 0-07-037996-3.



2. Grady Booch, *Análisis y diseño Orientado a Objetos. Con Aplicaciones*. Segunda Edición, Addison Wesley, 1996.
3. Rebeca Wirfs-Brock, Brian Wilkerson, Lauren Wiener. *Designing Object-Oriented Software*. Prentice-Hall, 1990.
4. Luis Joyanes Aguilar, *Programación Orientada a Objetos*, Mc Graw Hill, 1996, ISBN:84-481-0590-7
5. Bertrand Meyer. *Object-Oriented Software Construction*. Series in Computer Science. Prentice Hall, Englewood Cliffs, NJ, 1988.
6. J.J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, W. Lorensen, *Object Oriented Modeling and Design*. Prentice Hall Inc., 1991, New York (NY).
7. Gamma, Helm, Johnson, Vlissides, *Design Patterns. Elements of Reusable Objects Oriented Software*. Addison-Wesley, Professional Computing Series.

#### **Programación Orientada a Objetos en Smalltalk**

8. Wilf LaLonde. *Discovering Smalltalk*. Benjamin/Cummings, Redwood City, CA, 1994.
9. Wilf Lalonde & John Pugg. *Inside Smalltalk. Vol. 1*.
10. Wilf Lalonde & John Pugg. *Smalltalk V. Practice and Experience*.
11. Susanne Skublics, Edward J. Klimas, David A. Thomas. *Smalltalk with Style*, Prentice-Hall
12. Kent Beck. *Smalltalk Best Practice Patterns. Volume 1: Coding*.
13. Digital. *Smalltalk V. Object-Oriented Programming System*.
14. Dan Shafer, S. Herndon y L. Rozier. *Smalltalk Programming for Windows*.

#### **Lenguaje de Modelado Unificado (UML)**

15. The UML User Guide. Booch, Rumbaugh and Jacobson. Addison Wesley Longman, Inc, 1998.
16. The UML Reference Manual. Rumbaugh, Jacobson and Booch. Addison Wesley Longman, Inc, 1998.
17. UML gota a gota. Martin Fowler and Kendall Scott. Addison Wesley Longman de Mexico, 1999
18. Unified Modeling Language (UML) Specification - Version 1.4, September 2001. UML Specification, revised by the OMG, <http://www.omg.org>.
19. The Unified Software Development Process. Jacobson, I., Booch, G Rumbaugh, J., Addison Wesley. ISBN 0-201-57169-2 (1999)