

fiuba

algo3

Algoritmos y Programación III

Pablo Suárez

psuarez@fi.uba.ar

Ejercicio en grupos

Me dijeron que en Algoritmos III...



Algoritmos I

Introducción a la programación

Programación estructurada y sus estructuras de control

Estructuras de datos elementales

Algoritmos típicos

Modularización con subprogramas

Uso de archivos para almacenamiento fuera de memoria

Algún lenguaje de programación: Python y/o C

Algoritmos II

Estructuras de datos más complejas

Algoritmos más avanzados

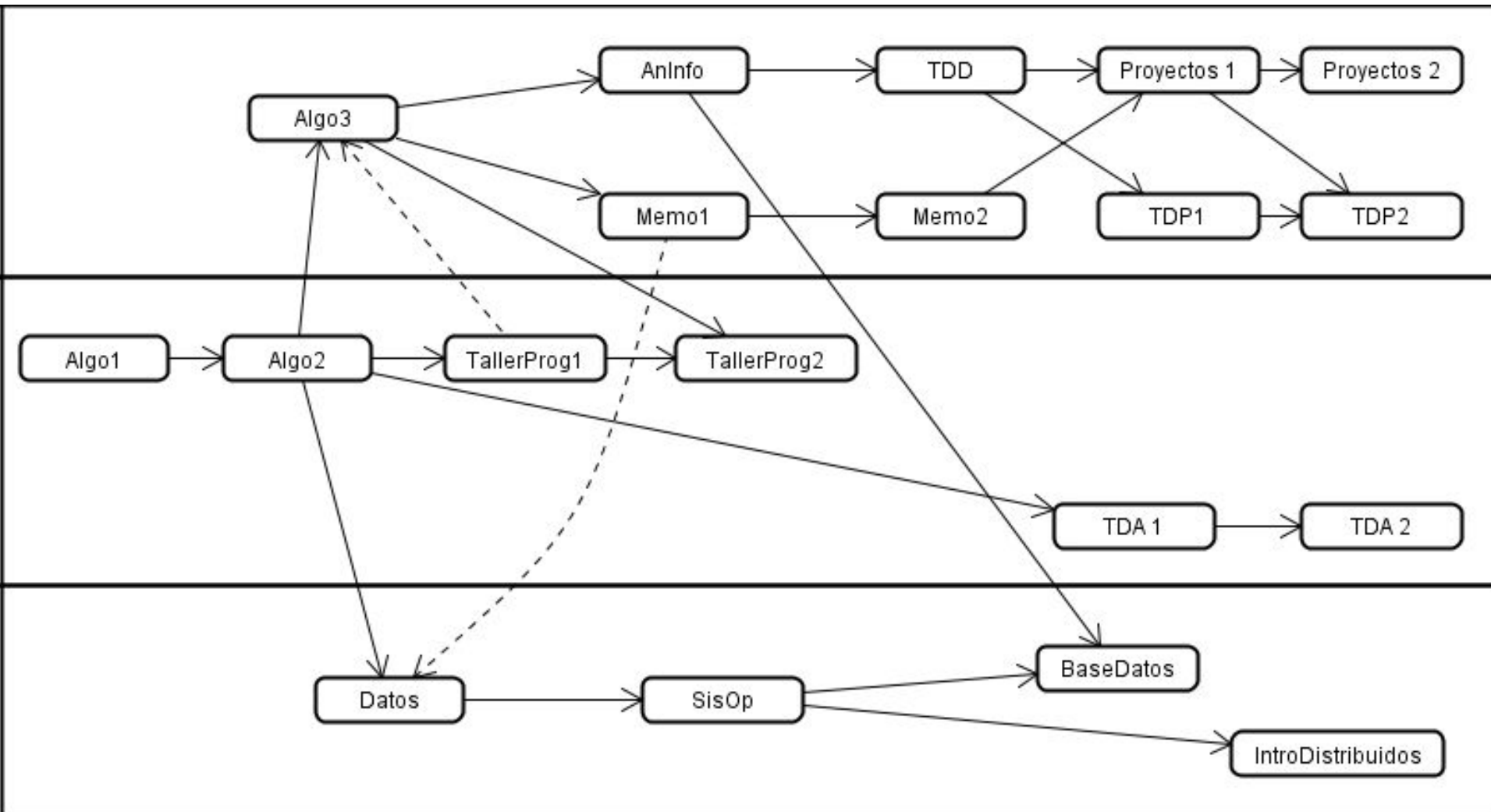
Nociones de complejidad de algoritmos

Modularización con tipos de datos definidos por el programador

Lenguajes Python y/o C y/o ++

=> Programas de mayor calidad

Materias afines y no tanto... (no oficial)



Temario Algoritmos III (1)

Programación orientada a objetos

Smalltalk, Java y otros

Calidad de código y diseño

Principios de diseño orientado a objetos

Cuestiones metodológicas

Test-Driven Development (TDD)

Refactorización

Modelado con UML

Temario Algoritmos III (2)

Temas alrededor de P00

Excepciones

Persistencia

Genericidad

Temas adicionales de programación superior

Pruebas

Interfaces de usuario y UX

Concurrencia



Objetivos de aprendizaje

Central

“Pensar en objetos”

Cuestiones metodológicas

Subsidiariamente

Leer literatura técnica

Comprender lo que se lee

Trasladarlo a la práctica

Explicar puntos de vista

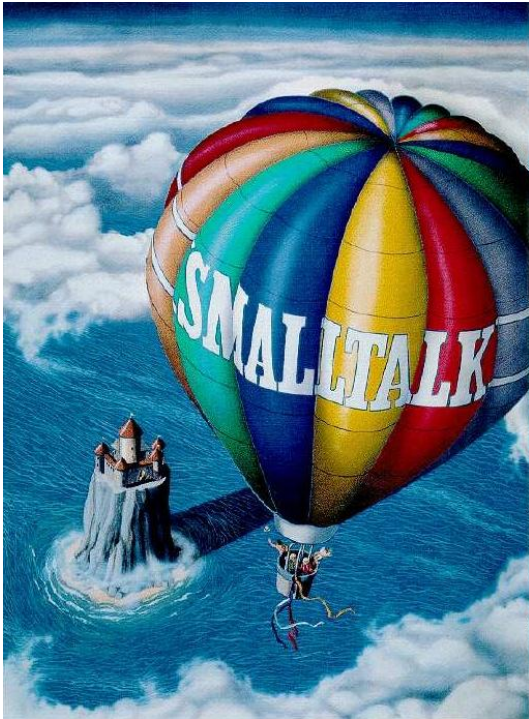
Oral: defensa último TP e integrador

Escrito: parciales

Aprender a aprender



Elecciones: ¿por qué?



Elecciones: visión de la cátedra

Smalltalk

Todo son objetos y mensajes, incluso el IDE, el compilador, etc. => modelo puro

Notación diferente a los derivados de C

Comprobación dinámica



Java

Gran difusión

Muchas herramientas

Comprobación estática

Notación derivada de C



UML

Estándar de facto para modelado

Importancia de modelar



Comunicación

Medio típico: Campus FIUBA

Material de clase se suben al campus

TPs, otros materiales obligatorios, anuncios, consultas

Twitter

@algo3fiuba

Sólo para avisos urgentes



Aprobación

1 parcial + integrador, con 3 oportunidades c/u

Temas teóricos, práctica conceptual

3 trabajos prácticos

1 de iniciación: Smalltalk (TP0)

1 individual de POO básica: Smalltalk (TP1)

1 en grupo avanzado y con IU: Java (TP 2)

2 evaluaciones sobre lecturas obligatorias

Exámenes cortos

Ver reglamento en el campus

Organización de la materia

Hay 2 cursos coordinados y con los mismos objetivos y teórica unificada

Martes a la noche y jueves a la tarde (TT - c1)

Martes y jueves a la noche (TN - c2)

Los integradores se toman en conjunto



Horarios y clases

Martes teórica y jueves práctica

Todas las clases son de asistencia obligatoria

Pretendemos participación

Aunque sean clases conceptuales

O precisamente porque lo son...

Clases prácticas

¡Prácticas!

Todo el que pueda traiga notebook



Bibliografía central

Esencial: diapositivas de las clases y lecturas obligatorias

Libro en elaboración de Carlos Fontela

Borrador en el campus

Soporte

Carlos Fontela, “Orientación a objetos con Java y UML”
(edición 2011)

Andrew P. Black, Stéphane Ducasse y otros, “Pharo Por Ejemplo”

Bruce Eckel, “Piensa en Java” o “Thinking in Java”

Martin Fowler, “UML gota a gota”

Opción: Carlos Fontela, “UML”

Recursos en la Web

Smalltalk y Pharo

<http://www.smalltalk.org/>

<http://www.pharo-project.org/home>

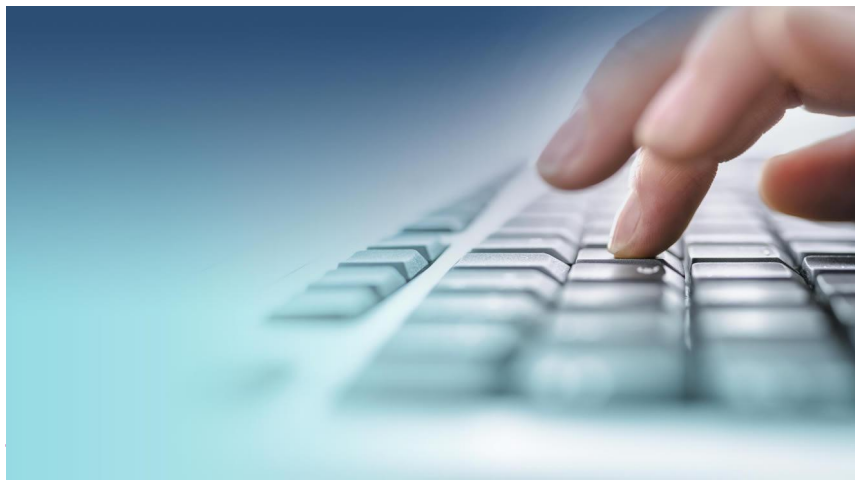
Java

<http://www.oracle.com/technetwork/java/>

UML

Varias herramientas

Otros: en cada tema



Docentes

Teórica:

Pablo Suárez, profesor

Lic. Sistemas FIUBA

También Profesor UNQui

director de carrera en UNQui



Docentes

Curso 1

Marcio Degiovannini, JTP

Ing. Informático FIUBA

También JTP en otra materia en FIUBA

Eugenio Yolis

Ing. Informático FIUBA

Estudiantes-ayudantes

Federico Jure

Edson Justo

Magalí Marijuán

Eugenia Franchi

Sebastián Blázquez

Camila Dvorkin



Docentes curso noche

Diego Sánchez

Ing. Informático FIUBA

También Profesor UNTREF

Estudiantes-ayudantes

Pablo Rodríguez Massuh

Tomás Bustamante

Martín Picco

Federico Jure



Próximos Pasos

Comenzar con la materia:

Resolución de problemas con objetos

Diseño por contrato

Como modelo de implementación de objetos

Colaboraciones de objetos y separación de responsabilidades

Incluye delegación, herencia
y cuestiones estructurales



Recapitulación



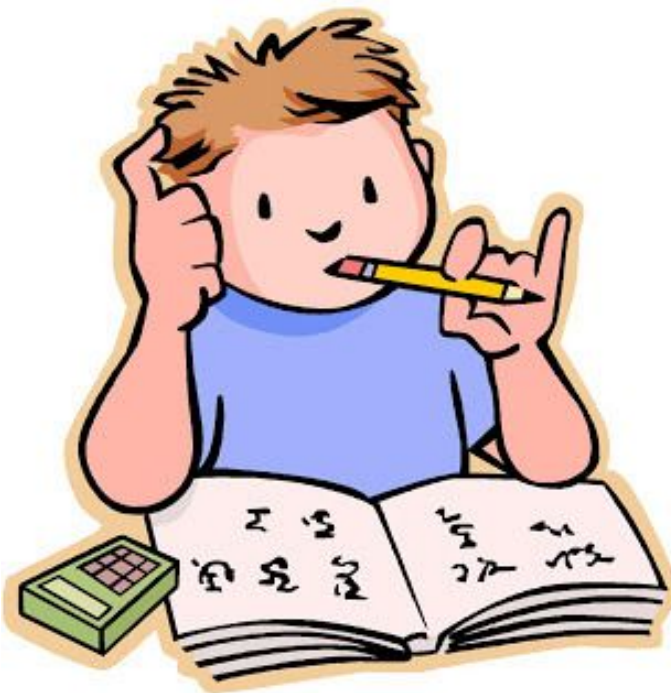
Recapitulación: preguntas

¿Cómo se aprueba la materia?

¿Cómo nos comunicamos?

¿Para qué hacemos esta recapitulación?

¿Para qué hicimos el ejercicio inicial?



Lectura interesante

Crash: how computers are setting us up for disaster

<https://www.theguardian.com/technology/2016/oct/11/crash-how-computers-are-setting-us-up-disaster>

(sí, es en inglés)