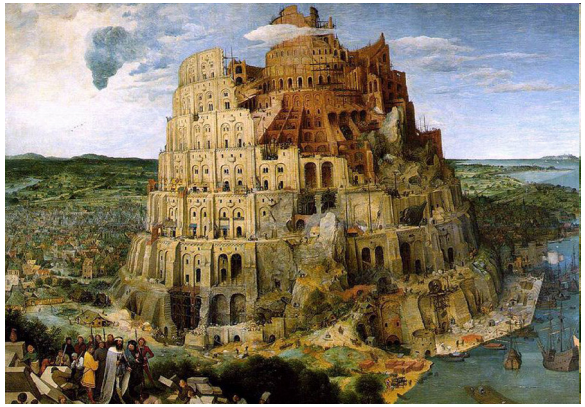
FACULTAD DE INGENIERIA
Universidad de Buenos Aires



Desarrollo de software y programación

Carlos Fontela
cfontela@fi.uba.ar

A3F

Temario

Desarrollo de software

Disciplinas del desarrollo

Programas y sistemas

Problemas de los proyectos de desarrollo de software

Desarrollo de software

- Principal ocupación de los egresados de carreras informáticas de FIUBA
  - Al menos los primeros años
- Desarrollo != Programación
  - Desarrollo de software incluye a la Programación
  - Pero también a otras disciplinas

Mini-historia del desarrollo

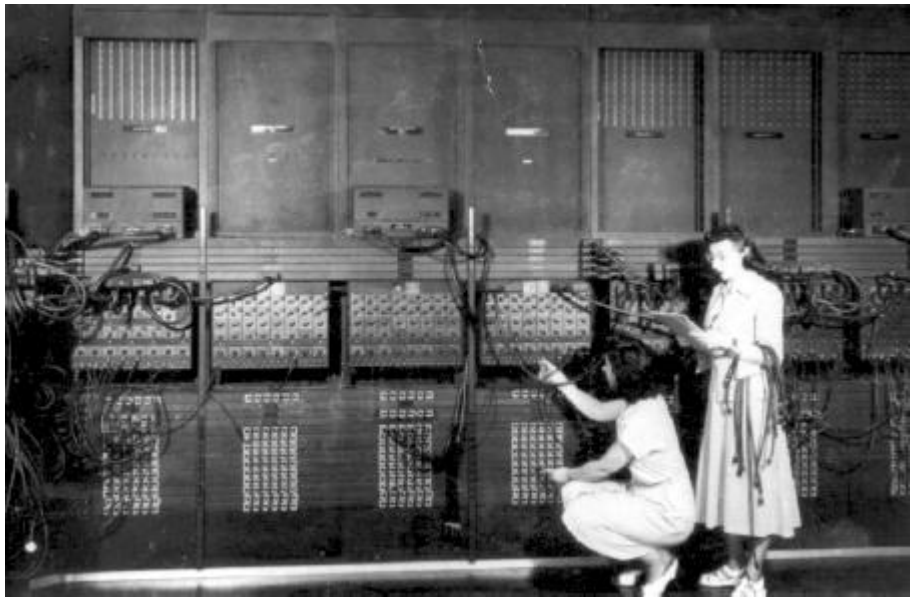
1945: ENIAC, programas cableados

1954: lenguajes de programación

1960s: aumenta la complejidad

1970s: sigue aumentando, búsqueda de un proceso, una "ingeniería"

> 1980: sigue aumentando, búsquedas en ingeniería y administración de proyectos



Mini-historia de carreras argentinas

- 1961: llega Clementina a la FCEN
- 1960s: carrera de Computador Científico en FCEN
  - Departamento de Matemática
  - Muy centrada en programación
- 1970s: carrera de Analista Universitario de Sistemas en FIUBA
  - Centrada en problemas ingenieriles y cuestiones organizacionales
- 1990s: carrera de Ingeniería Informática en la FIUBA
  - Y cambio de nombres de las carreras antiguas: licenciaturas
- 1990s: primera Facultad de Informática de la Argentina: UNLP

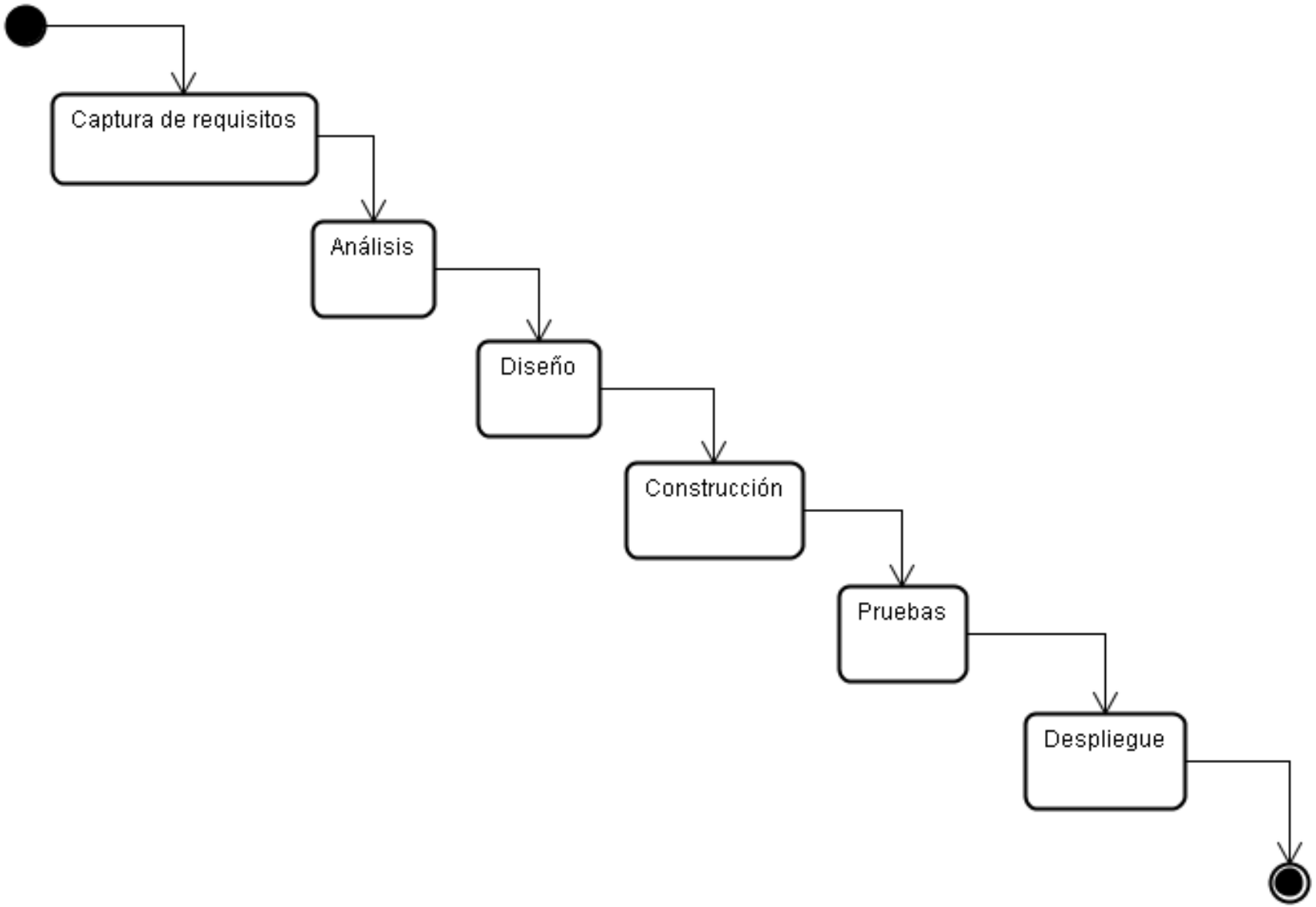
Disciplinas de desarrollo de software

No todo es programación en el desarrollo de software

Muchas visiones

En la década del 70 se definieron, asociadas a un método, hoy llamado “cascada”

Desarrollo en cascada



Disciplinas del desarrollo (operativas)

Captura de requisitos: qué quiere el cliente

Análisis: qué vamos a construir

Diseño: cómo

Implementación o construcción

Pruebas: verificación y validación

Despliegue: la hora de la verdad

Disciplinas del desarrollo (soporte)

Administración del proyecto, incluyendo seguimiento y control

Gestión de cambios

Administración de la configuración

Gestión de los recursos humanos

Gestión del ambiente de trabajo

Gestión de la calidad

Captura de requisitos

Muy difícil

Nunca se puede anticipar totalmente la funcionalidad de un producto

Cliente: “No sé lo que quiero, pero si me mostrás algo, te digo por qué no me gusta”

Requisitos != Expectativas != Necesidades

No se puede congelar requisitos ni tan siquiera en un proyecto de mediano porte

Análisis

Sistema a construir

- Incluye cuestiones tecnológicas que sean requisitos

Requisitos != Expectativas != Necesidades

- Tratar de concentrarse en las necesidades

Se obtienen especificaciones funcionales de la aplicación

Se deben acordar con el cliente

Diseño

Actividad eminentemente ingenieril

- Determinamos cómo resolver el problema

Muchos aspectos técnicos

Algo veremos en Algoritmos III

Hay materia específica

Pruebas

Validación

- Que el sistema haga lo que el cliente espera

Verificación

- Que el sistema haga lo que dicen las especificaciones

Hay muchos tipos de pruebas

- Complementarias

Tema de Algoritmos III y otras materias

Disciplinas y cascada

No siempre están atadas

Ni tienen que estarlo

Es un tema que abordaremos luego, finalizando la materia

Programas y sistemas

Sistema != Programa

tanto como Desarrollo != Programación

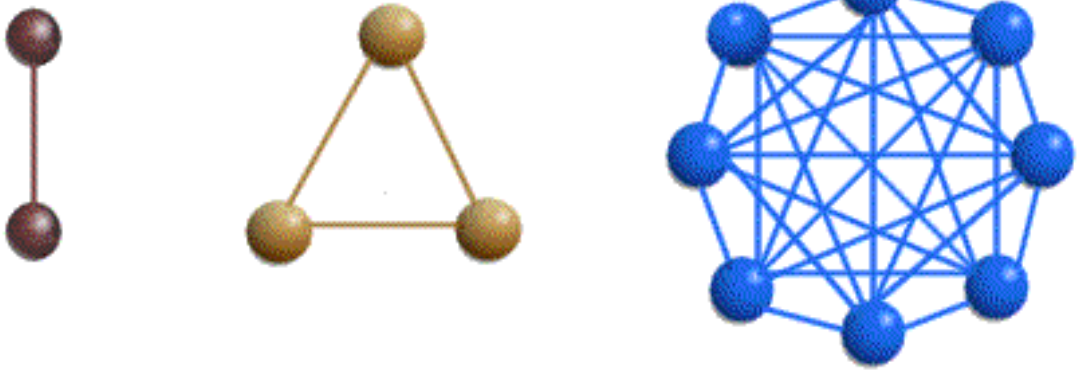
Sistema:

Conjunto de módulos interrelacionados

Puede no haber un “programa principal”

Pueden ser muchos programas comunicándose

Complejidad aumenta con el número de partes



Software y hardware

Software =

Programas +
Datos +
Documentación +
Conocimiento y reglas del dominio del problema

El software controla al hardware

Y lo hace útil

Cada vez hay más aparatos controlados por software

Y cada vez más funciones del aparato

Características del software (1)

- Intangible
- Maleable ("soft")
  - No necesariamente implica facilidad de cambio
  - Sí posibilidad
- Se desarrolla por proyectos
  - Diferencia con otros productos industriales
  - Parecidos con la industria de la construcción
- Alto contenido intelectual
  - Generalmente disperso y difícil de reunir
  - Diseñado y construido por profesionales

Características del software (2)

¿Producción de software?

- Si hay "producción" de música y películas…
- Es una actividad humana

El software no se "fabrica"

- Se **construye** o se **desarrolla**

Mantenimiento constante

- Desde su construcción
- "Como con algunos parientes, es difícil deshacerse de algunos productos de software" (Freeman)

Fracasos del desarrollo de software (1)

- Proyectos que no terminan a tiempo
  - Aeropuerto de Denver: sistema de administración de equipajes
  - agosto 1994 => diciembre => marzo => mayo
  - pérdidas de U$S 1 M por día de atraso
- Proyectos que cuestan más de lo estimado
  - PPARS (proyecto de administración de personal, Irlanda)
  - Estimado en € 8,8 M => € 140 M

Fracasos del desarrollo de software (2)

- "Accidentes"
  - Software del Ariane 5
    - Explota a poco de salir por pérdida total de información de guiado y altitud
    - Origen: uso de software del Ariane 4
  - Accidente con pacientes oncológicos en Panamá
- Productos que no cumplen lo que el solicitante quiere
  - 45% del software contratado nunca es usado

Problemas en otras ingenierías de proyectos (1)

Tacoma – Narrows

- Colapsó en 1940

Puente de Aviñón

- Construido en 1171
- Destruido varias veces
- Último intento 1660

Problemas en otras ingenierías de proyectos (2)

- Accidentes en Three-Mile Island y Chernobyl
- Yacyretá-Apipé
  - US$ 11.000 M en 15 años
- Puente Chaco-Corrientes
  - Fallas de diseño (1973)
- Big-Dig, Boston, EEUU
  - 2,8 MM => 14 MM
- Catedral de Colonia, Alemania
  - 1248-1880
  - Reparación permanente

2 casos propios

1987

- Tecnologías desconocidas
- Poca experiencia en desarrollo
- Pruebas al final
- 1 mes => 3 meses

2007 (¡20 años más tarde!)

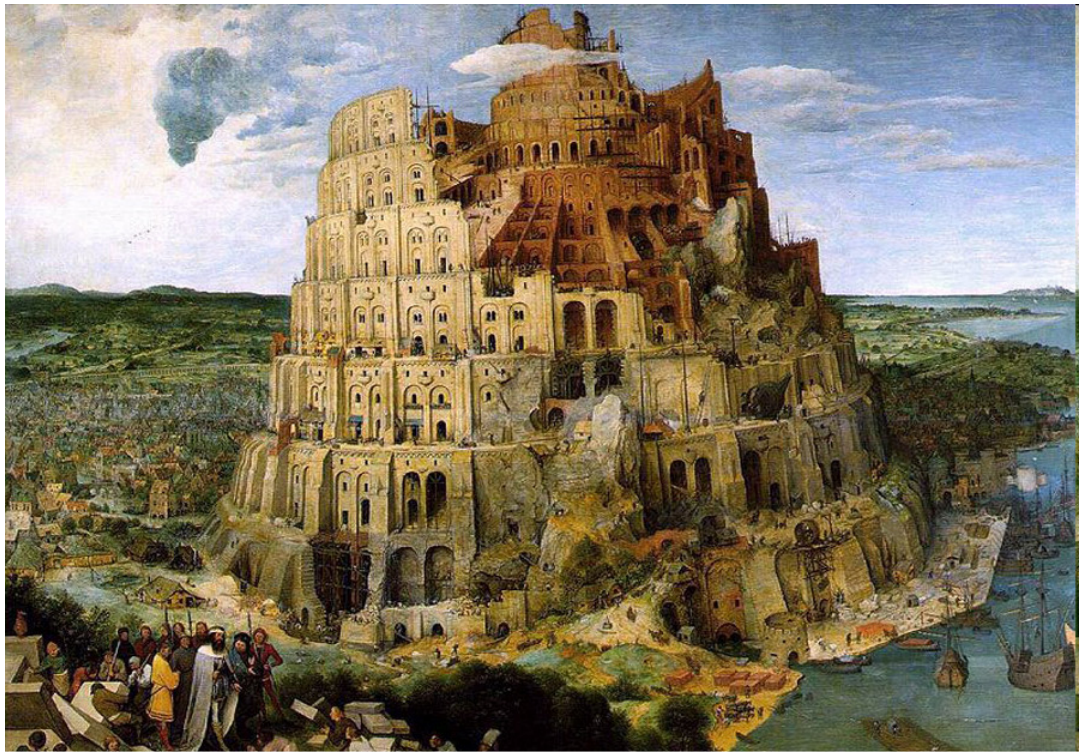
- Fecha de entrega fija
- Demoras en comienzo
- Interacción con otro proveedor
- Problemas de equipo
- 2 meses => 6 meses

Algunas reflexiones

Los problemas del desarrollo no son sólo tecnológicos

Ley de Brooks: agregar gente a un proyecto atrasado lo atrasa más

Cuidar la comunicación



Desarrollo y Algoritmos III

Es una materia de Programación

- La última obligatoria de contenidos

Pero también se ve

- Diseño
- Pruebas
- Buenas prácticas metodológicas
- Buena calidad de código
- Usabilidad

Claves

Desarrollo >> Programación

Sistemas >> Programas

Problemas desarrollo >> Problemas tecnológicos

Lectura obligatoria

Repasar abstracción y ocultamiento de Algoritmos II

"Tipos de datos abstractos" => "Tipos definidos por el programador"

"Ocultamiento de la implementación"

Lecturas opcionales

Artículo de Wayt Gibbs en Scientific American, "Software's Chronic Crisis" en: http://www.cis.gsu.edu/~mmoore/CIS3300/handouts/SciAmSept1994.html

Paper de Fred Brooks, "The Mythical Mon-Month": buscar en la Web

Artículos de Carlos Fontela en blog: http://cysingsoft.wordpress.com/

Qué sigue

Objetos (uso)

Clases (construcción)

Delegación y herencia