

**UNIVERSIDAD DE BUENOS AIRES**  
**FACULTAD DE CIENCIAS EXACTAS Y NATURALES**

1. DEPARTAMENTO: Computación
2. CUATRIMESTRE: Segundo de 2004.
3. ASIGNATURA: **Avances en el Testeo de Sistemas Complejos**
4. CARRERA: Licenciatura en Ciencias de la Computación
5. CARACTER DE LA MATERIA: Optativa
6. NUMERO DE CODIGO DE CARRERA: 18
7. NUMERO DE CODIGO DE MATERIA:
8. PUNTAJE: 1 punto
9. PLAN DE ESTUDIOS AÑO: 1993
10. DURACION: 1 semana
11. HORAS DE CLASE SEMANAL:  
a)TEORICAS/PRACTICAS: 15 horas    b)LABORATORIO:                    c)PRACTICAS:
12. CARGA HORARIA TOTAL: 15 horas
13. ASIGNATURAS CORRELATIVAS: **Conocimientos básicos de 1) álgebra (matrices, permutaciones, estructuras) 2) algoritmos y estructuras de datos**
14. FORMA DE EVALUACION: final
15. PROGRAMA Y BIBLIOGRAFIA:

Profesor  
**Dr. Sebastián Elbaum**

  
Dr. Enrique Carlos Segura  
Director  
Depto. de Computación  
F. C. E. y N - UBA

## 15) PROGRAMA:

**Introducción:** Un estudio reciente realizado por el National Institute of Standards and Technology (NIST-USA-2003) reportó que fallas en software cuestan a los Estados Unidos al menos \$59 mil millones cada año, y estimo que el uso de técnicas de testeo apropiadas podrían ahorrar \$22 mil millones. El testeo de software es definitivamente una actividad fundamental para el desarrollo de grandes sistemas que sean robustos y confiables. Sin embargo, el aumento de la complejidad, los niveles de configuración, y el numero de usuarios de los sistemas modernos limita la efectividad de técnicas de testeo tradicionales. Por ejemplo, Microsoft XP tiene aproximadamente 50 millones de líneas de código, cientos de librerías compartidas, y millones de usuarios que exponen el software a situaciones impredecibles. Es irrazonable suponer que los ingenieros logren validar con efectividad un sistema de esta naturaleza. Al mismo tiempo, los crecientes niveles de conectividad y el poder de nuevas técnicas de análisis dinámico que operan durante la ejecución de un programa abren nuevas oportunidades para agilizar y hacer mas efectiva la actividad de testeo.

**Objetivos:** Este curso tiene como objetivo rever técnicas de testeo, introducir técnicas de captura y análisis de datos durante la ejecución de un programa, y presentar recientes aplicaciones prácticas en el área de testing.

### Programa: 1- Presentación del curso

- Expectativas y objetivo
  - Motivación y oportunidades
- Introducción a técnicas de testeo**
- Principios, fundamentos y terminología
  - Estrategias sistemáticas, límites y particiones
- Derivando modelos de programas**
- Derivaciones de modelos formales de estado
  - Diagramas de dependencia de control
- Diagramas de dependencia de datos**

### 2- Estrategias de testeo en base a la estructura de programas

- Criterios de cubrimiento
  - Mutación de programas
- Aplicaciones**
- Estimación de impacto de cambios
  - Testeando programas que evolucionan

### 3- Técnicas para análisis dinámico: precisión y eficiencia

- Captura de datos durante la ejecución del programa
  - Análisis de la estructura de programas para reducir tiempo de ejecución
  - Estrategias de muestreo para agilizar la ejecución
- Aplicaciones**
- Perfiles de ejecución de un programa
  - Detectando invariantes automáticamente

### 4- Monitoreando y testeando sistemas después de distribución

- Distribución de instrumentación
  - Muestreo de instrumentación, funcionalidad y usuarios
  - Detección de anomalías durante la ejecución
- Aplicaciones**
- Generación automática de tests
- Detección de comportamientos inesperados**


### 5- Enfoques prometedores

  
Dr. Enrique Carlos Segura  
Director  
Depto. de Computación  
F. C. E. y N. UBA

**Impacto en varios tipos de aplicaciones**  
**Sesión abierta: repaso de lecciones más importantes**  
**Conclusiones**

16) BIBLIOGRAFIA:

Apunte de cátedra. No fue adjuntada otra bibliografía por parte del docente a cargo.

  
Dr. Enrique Carlos Segura  
Director  
Depto. de Computación  
F. C. E. y N - UBA