

**Departamento de Informática – Facultad Politécnica
Ingeniería Informática**

ALGORITMOS Y ESTRUCTURA DE DATOS III

2018 – 1er. Semestre – Sección TQ

Prof. Cristian Cappel
ccappo@pol.una.py
ccappopy@gmail.com

AER: Cristian Aceval (a confirmar)

Algoritmos y ED III

- **Horario de clases (Aula ???)**

- Miércoles de 17:30 a 19:45 (2,25 HR)
- Jueves de 14:30 a 17:30 (3HR)

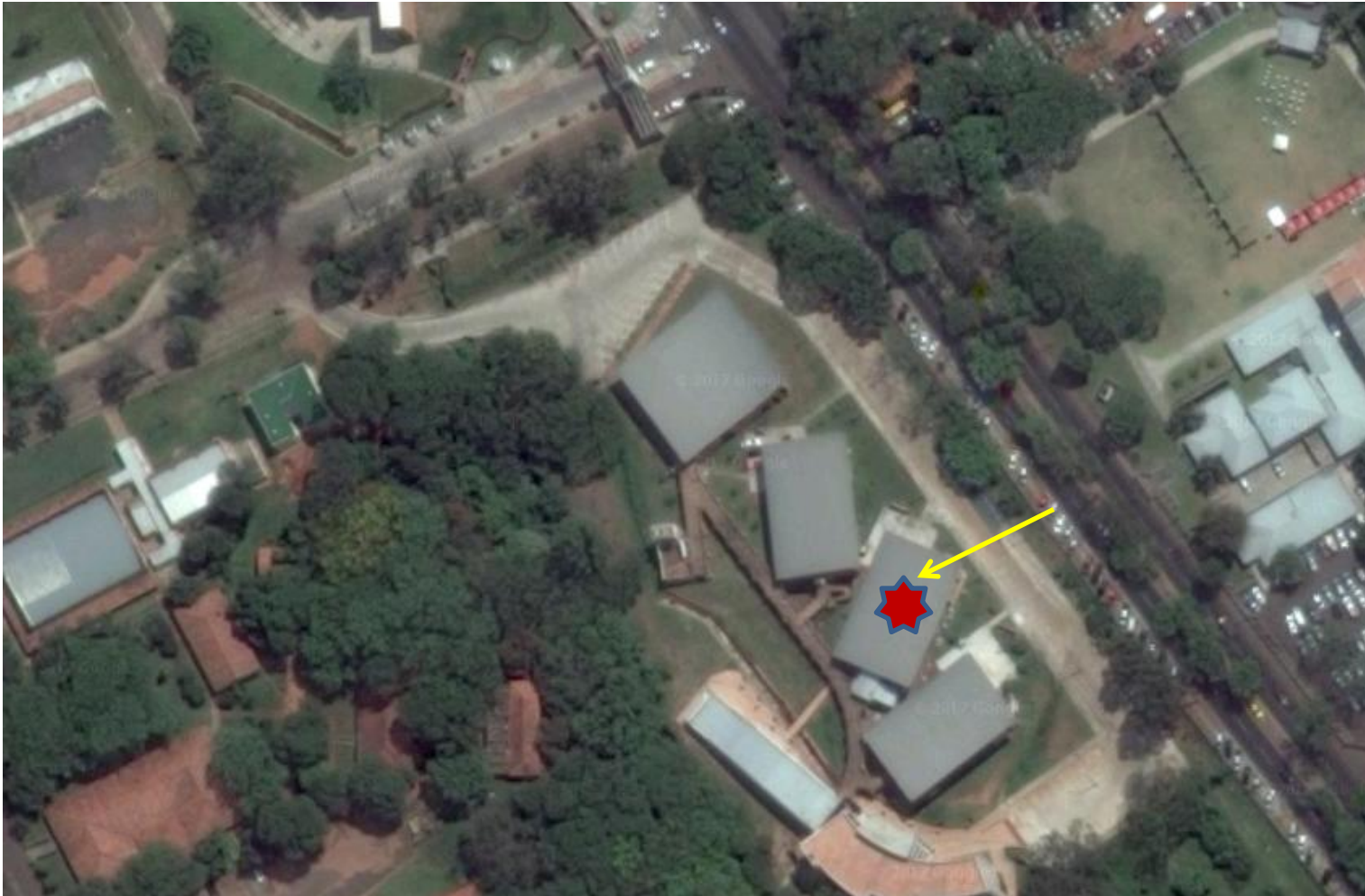
POR FAVOR LLEGAR A HORA A LA CLASE. Los alumnos sin DEF(*Derecho a Examen Final*) deben asistir al menos el 75% de las clases.

- **Consultas extra-clase:**

- Sin problemas, vía correo electrónico
- Coordinando una visita a la oficina del profesor. **Local:** *Núcleo de Investigación y Desarrollo Tecnológico (NIDTEC) de la FPUNA.* Bloque 3 del exCET (bloque de la FPUNA), en la entrada del campus sobre la Avda. Mcal. Lopez.

Donde ubicarme

(Usualmente de 8:00 a 18:00 hs,
aunque mejor confirmar por mail ☺)



¿Que haremos en el curso?

- Estudiar algoritmos conocidos y de gran influencia
- Estructura de datos (ED) avanzadas.
- Examinar métodos para analizar el rendimiento (*eficiencia*) de algoritmos
- Aplicar criterios para selección de algoritmos y ED según la naturaleza del problema a resolver.
- Construir código claro, depurable y documentado (*aunque no sea un curso de programación ni de ingeniería de software*)

Objetivos del curso

ver [Programa de estudio](#) (revisar)

- Evaluar críticamente la aplicabilidad de un algoritmo ó estructura de dato en el contexto de problemas concretos.
- Comprender e implementar los algoritmos fundamentales en Computación.
- Evaluar el costo/beneficio que existen entre varios algoritmos que ofrecen la misma funcionalidad.
- Aplicar diferentes técnicas de diseño de algoritmos para resolver problemas computacionales.
- Emplear correctamente las técnicas de análisis de algoritmos como herramienta fundamental en la toma de decisiones de diseño e implementación de soluciones algorítmicas a problemas computacionales.

Contenido

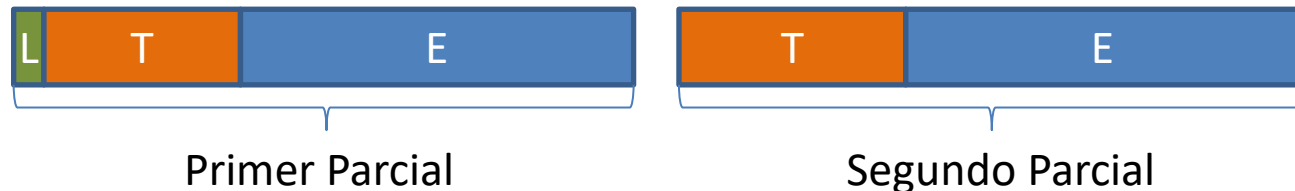
1. Java (*como herramienta auxiliar y transversal*).
Un uso desde el punto de vista de Ingeniería de SW verán en otras asignaturas.
1. Análisis de algoritmos (***unos de los temas más importantes y de carácter transversal***)
2. Ordenación y búsqueda
 - Árboles binarios (no balanceados (BST) y balanceados (AVL, Rojinegros, etc))
 - Tablas de dispersión
 - Ordenación interna (##Sort basados o no en comparación)
 - Ordenación externa (ej: B-Arboles)
3. Búsqueda de patrones en cadenas (KMP, Boyer-Moore, Rabin-Karp)
4. Grafos (*ED y Algoritmos más relevantes: barrido, búsqueda, cubrimiento, etc*)
5. Técnicas de diseño de algoritmos :
 - Divide y vencerás
 - Algoritmos voraces o greedy,
 - Programación Dinámica,
 - Backtracking o vuelta atrás.
6. Algoritmos aleatorizados o probabilísticos (análisis probabilístico, ED con base aleatoria)

¿Qué veremos?

- Estudiaremos fundamentalmente “**Análisis de Algoritmos**” (*es el estudio teórico del rendimiento de programas y el uso de recursos*)
- Otros factores, son también muy importantes:
 - Modularidad
 - Correctitud
 - Mantenibilidad
 - Seguridad
 - Funcionalidad
 - Robustez
 - Usabilidad
 - Simplicidad
 - Extensibilidad
 - Escalabilidad
 - Confiabilidad

Reglas del curso (1)

- El curso será teórico/práctico. Mucha programación 😊
- Se tendrán tareas semanales (10 en total)
- El puntaje parcial (100%) será dividido de la siguiente forma:
 - 60% **Examen (en el 2do parcial será 65%)**
 - 35%
 - 05% **Laboratorio (no hay en el período del 2do parcial)**



- Si no rinde el primer o segundo parcial, el puntaje de tareas y laboratorio correspondiente a ese parcial se suman al tercer parcial. Si rinde ambos exámenes y no consiguió derecho, el tercer examen parcial es 100%.

Reglas del curso (2)

- El nro. de tareas será de 10 distribuidas en todo el curso. Muchas de ellas son de programación (esencialmente en Java), aunque también habrá mucha teoría.
- Las tareas se entregan por grupo de 1 a 2 personas vía EDUCA (www.educa.una.py/politecnica). Favor colocar una foto actual en su perfil de EDUCA. Enviar el nombre de los integrantes del grupo vía el foro FAQs del curso en EDUCA (hasta este miércoles 21/FEB). La clave de acceso es “alg3_2018_Stroustrup” sin las comillas.
- Los documentos, ejercitatorios, presentaciones, etc, se mantienen en el sitio de EDUCA.
- Los exámenes serán SIN material.

Recomendaciones



- Dedicarle al menos 5 horas a la semana para estudiar, leer y practicar.
- Leer por anticipado, para ello tiene a mano el plan semestral con las referencias de lectura. **Use el Plan Semestral.**
- Participar y consultar en clase.
- Usar EDUCA para discutir temas que no quedaron claros o aquellos relacionados a los temas del curso.

El profesor podría requerir algún tiempo en dar la respuesta, en caso de que no la tenga a mano.

Plan Semestral (20 clases en total con 10 tareas y 2 laboratorios)

TEMA/Actividad	Fechas	Entrega TAREA
Introducción + Java + Laboratorio Java	15/FEB, 21/FEB, 22/FEB(L)	1,2 (<i>Java y C</i>)
Análisis de Algoritmos	28/FEB y 1/MAR	3 (<i>Análisis de alg.</i>)
Ordenación y Búsqueda: BST + Laboratorio BST	7/MAR y 8/MAR(L)	4 (<i>BST</i>)
Ordenación y Búsqueda: AVL	14/MAR	
Ordenación y Búsqueda: Tablas de dispersión	15/MAR	5 (<i>Tablas Hash</i>)
Ordenación y Búsqueda: Ordenación interna	22/MAR	6 (<i>Ord. Interna</i>)
Primer Parcial (13/ABR)	2/ABR - 14/ABR	Tareas: 1,2,3,4,5
Ordenación y Búsqueda: Ordenación Externa	18/ABR	7 (<i>Ord. Externa</i>)
Búsqueda de Patrones	19/ABR	
Grafos	25 y 26/ABR, 2-3/MAY	8 y 9 (<i>Grafos</i>)
Técnicas de diseño de Algoritmos	9 y 10/MAY	10 (<i>Tec. Diseño Alg.</i>)
Algoritmos aleatorizados	16 y 17/MAY	
Segundo Parcial (01/JUN)	19/MAY – 01/JUN	Tareas: 6,7,8,9,10
Primer Final (22/JUN)	08/JUN – 23/JUN	
Segundo Final (06/JUL)	25/JUN – 09/JUL	

Bibliografía

- **Estructura de Datos en Java**, Mark Allen Weiss. 1998.[WEIS2000]
- **Data structures and algorithm analysis. Edition 3.2 Java Edition**. Clifford A. Shaffer. 2013. [Shaffer2013] (*disponible en línea*)
<http://people.cs.vt.edu/~shaffer/Book/JAVA3elatest.pdf>
- **Introduction to Algorithm**. Cormen, Leiserson, Rivest & Stein. 2nd Edition. 2001. [CLRS2001] ó 3rd. Edition. 2009 [CLRS2009].
- **Algorithms**. S. Dasgupta, C. Papadimitriou & U. Vazirani . 2008. McGraw-Hill.
- **Algorithms**. R. Sedgewick & K. Wayne. 4th Edition. 2011. Addison-Wesley.
- **Data Structures and Algorithm Analysis in Java**. Mark. A. Weiss. 3rd. Edition. Pearson. 2012.
- **The introduction to the design and analysis of algorithms**. 2nd Edition. Anany Levitin. 2007.
- **The algorithm Design Manual**. Steven Skiena. 2nd edition. 2008. Springer
- **Algorithm Design**. Jon Kleinberg & Eva Tardos. 1th Edition. 2006. Pearson Education.
- **Estructura de datos y algoritmos**. Aho, Hopcroft & Ullman. Addison-Wesley. 1988
- **Algoritmos y estructura de datos**. Niklaus Wirth. Prentice-Hall. 1987
- **Estructura de datos y algoritmos**. Mark Allen Weiss. Addison-Wesley. 1995
- Artículos y documentos adicionales entregados por el profesor (ver Referencias en EDUCA).

¿Preguntas hasta aquí?

Empecemos

Conceptos básicos

Problema

Propiedades de un
algoritmo

Algoritmo

Tipo

Tipo Abstracto de Dato

Tipo de dato

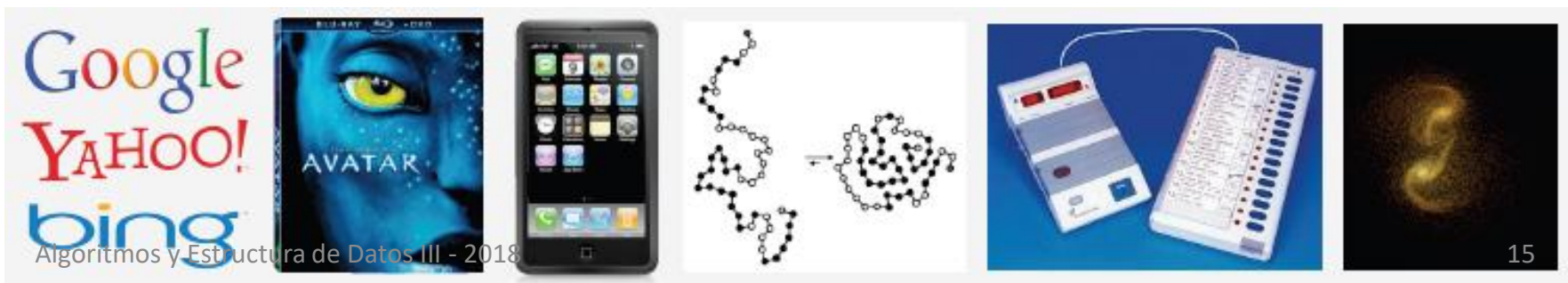
Estructura de datos

Algoritmo eficiente

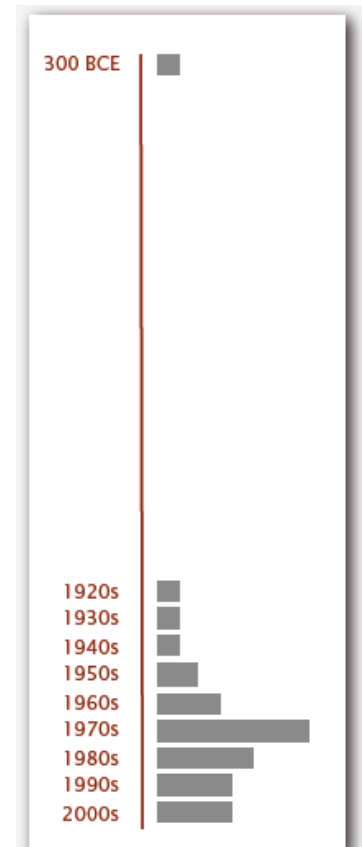
Lenguaje de programación

¿Porqué estudiar algoritmos y ED?

- **Internet:** web search, ruteo de paquetes, archivos distribuidos, cloud computing, IoT, ..
- **Biología:** proyecto del genoma humano, plegamiento de proteínas, ..
- **Computación:** circuitos, sistema de archivos, compiladores, ..
- **Computación gráfica:** películas, video games, realidad virtual, 3D, ..
- **Seguridad:** web-security, e-commerce, e-voting, ..
- **Multimedia:** MP3, JPG, DivX, HDTV, reconocimiento de rostros, ...
- **Redes sociales:** sistemas recomendadores, propaganda, noticias, ..
- **Física/Química:** simulación de fenómenos físicos o químicos (colisión de partículas, astronomía, procesos químicos, cosmología, etc)



- **Algoritmo - Origen:** del nombre del matemático y astrónomo árabe “**Abu Ja'far Mohammed ibn Musa al-Khwarizmi (825)**” (*Padre de Ja'far, Mohammed, hijo de Moisés, natural de Khowarizm*). Escribió el célebre libro “*Kitab al jabr w'al-muqabala*” (*Reglas de restauración y reducción -- > “álgebra”*)
- Viejas raíces, nuevas oportunidades
 - El estudio de los algoritmos data al menos de la época de Euclides
 - Formalizado por Church y A. Turing en 1930



Solución de problemas

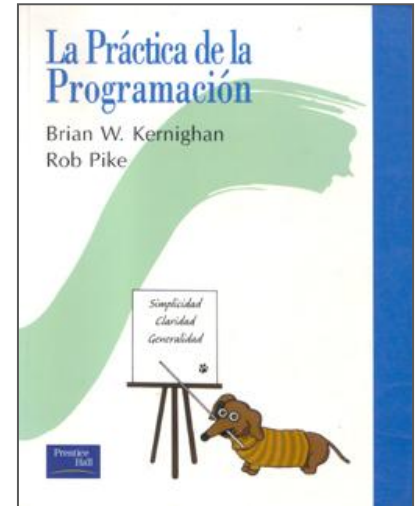


¿Porqué estudiar algoritmos?

- **Comprender la naturaleza intrínseca del problema, así como las posibles técnicas de solución independientes al lenguaje de programación, paradigma, hardware u otro aspecto de implementación.**
- Es transversal y de largo alcance (base fundamental para otras áreas de computación)
- Para resolver problemas difíciles que se nos presenta
- Para ser mejor y un proficiente programador
- Por una estimulación intelectual
(*Por ejemplo, competencias de programación – ACM-ICPC : <http://icpc.baylor.edu>)*
- Divertido 😊
- Generador de buena\$ oportunidad\$ en cualquier área

Para escribir un programa

- Utilice un estilo de programación (leer el libro *La práctica de la programación* de Brian Kernighan y Rob Pike):
 - Nombre “conveniente” a variables
 - Ser consistente (del mismo modo en todos los lugares)
 - Nombre activo para funciones
 - Ser preciso (nombre vs funcionalidad)
 - Usar sangrías (adecuadamente)
 - Dividir expresiones complejas
 - Usar forma natural en las expresiones (no escriba código oscuro)
 - Emplear convenciones (por ejemplo el *Java Convention Code* - JCC)
 - Dar nombre a números mágicos
 - Comentarios en código no obvio. Aclarar, no confundir.
 - Funciones y datos globales siempre deben comentarse
 - No comentar código malo, debe reescribirlo.
 - Y hay más .. mucho más



¿Para qué tomarse la molestia?

- ¿Porqué preocuparse por el estilo?
- ¿A quién le importa cómo se vea un programa si éste funciona?
- ¿No toma demasiado tiempo hacer que se vea “bonito”?



- El código bien escrito, ordenado y con estilo, es más fácil de leer y entender. Y posiblemente tiene menos errores.
- El enemigo del programador: el tiempo. Pero no es excusa.
- El código desaliñado es malo, torpe, difícil de leer y muchas veces hasta incorrecto.
- EL BUEN ESTILO DEBE SER UN HÁBITO.

Para las próximas clases

- **Miércoles 21/FEB/2018**

- Haberse inscripto en Educa / Enviar integrantes de grupo al Foro de Educa en el tema publicado por el profesor.
- Básicamente leer: *semejanzas y diferencias con lenguaje C*.
 - Tutorial de Java en el sitio de Oracle:
<http://docs.oracle.com/javase/tutorial> (mirar los tópicos: *Getting Started*, *Learning the Java Language* y si se puede: *Essential Java Classes*)
 - Para practicar y aprender Java de una forma sencilla, puede mirar el sitio:
<http://codingbat.com>
 - En este sitio encontrarán información paso a paso de como utilizar Java para programar (que es del libro “*Introduction to Programming in Java – An interdisciplinary approach*”) .
<http://introcs.cs.princeton.edu/java/home/>
 - Utilizar/revisar enlaces puestos en el Plan Semestral.

- **Jueves 22/FEB/2018 (via Educa)**

- Tarea #1, Ejercicios relacionados a TDA y lenguaje C.
Ya se encuentra disponible en EDUCA.

¿Preguntas/Comentarios?