

**Formulario para la presentación de Cursos de Posgrado/Doctorado - Res.  
CD2819/18 - ANEXO 1**

**Información académica**

Año de presentación (\*)

2021

1-a-

Departamento docente que inicia el tramite:
COMPUTACION
Nombre del curso:
Modelos, Algoritmos y Programación
Nombre, Cargo y Título del docente responsable:
Francisco Soullignac, Profesor Adjunto DE, Dr. de la UBA área Cs. de la Computación
En caso de dictarse en paralelo con una materia de grado, nombre de la misma:
Problemas, Algoritmos y Programación
Nombre y Título de los docentes que colaboran con el dictado del curso (*) (*):
Ayudante de primera o segunda a designar
Fecha propuesta para el primer dictado luego de la aprobación:
1er cuatrimestre 2021

**Duración:**

Duración total en horas	96 horas
Duración en semanas	16 semanas

Distribución carga horaria:

Número de horas de clases teóricas	32
Número de horas de clases de problemas	32
Número de horas de trabajos de laboratorio	32
Número de horas de trabajo de campo	
Número de horas de seminarios	

**Forma de evaluación:**

12 trabajos prácticos semanales con implementaciones computacionales de diferentes problemas. Final teórico

Lugar propuesto para el dictado (departamento, laboratorio, campo, etc.):

Departamento de Computación

Puntaje propuesto para la carrera de doctorado:

4 puntos

Número de

Mínimo: 1

Máximo: 20

alumnos:		
----------	--	--

**Audiencia a quien está dirigido el curso:**

Graduados de carreras de computación. Graduados de otras carreras que cuenten con conocimientos avanzados de programación y conocimientos básicos de matemática discreta.

**Necesidades materiales del curso:**

Versión presencial: aula, pizarrón y proyector.

Versión a distancia: sistema de videoconferencia (e.g., zoom, meet o jitsi)

1-b-

**Programa analítico del curso con Bibliografía (puede adjuntarse en hojas separadas):**

Implementación:

- Implementación en lenguajes imperativos/de objetos (C++, Python o Java)
- Reuso de herramientas y bibliotecas del lenguaje (estructuras y algoritmos)
- Estrategias de implementación

Técnicas de programación y algoritmos generales:

- Backtracking
- Dividir y conquistar
- Programación dinámica / recursión memoizada
- Golosos
- Búsqueda binaria y ternaria
- Problemas de transformación de estados y simulación

Teoría de grafos y algoritmos sobre grafos

- Propiedades generales de grafos y de clases de grafos particulares
- Problemas de grafos conocidos: caminos mínimos, árbol generador, matching, flujo, etc
- Modelado de problemas con grafos

Estructuras de datos básicas y avanzadas:

- Repaso de trie, AVL, patricia, heap y otras estructuras para diccionarios
- Suffix tree, suffix array, RMQ tree, árbol de mínimos, etc
- Formas de representación y sus propiedades
- Estructuras sobre punteros

Aritmética para resolución de problemas:

- Aritmética entera: divisibilidad, primos, factorización, mcm, mcd, congruencia, etc
- Aritmética en Polinomios, Matrices, etc
- Combinatoria
- Probabilidades, usos de la mediana

Geometría para la resolución de problemas:

- Distintos espacios ( $\mathbb{R}^n$ , grillas, superficies)

- Trigonometría con punto flotante en aritmética finita
- Geometría proyectiva con enteros (puntos, planos, rectas).
- Construcciones geométricas en la computadora
- Operaciones con figuras geométricas elementales: polígonos, circunferencias, poliedros, etc

Algoritmos para cadenas:

- Matching perfecto
- Expresiones regulares
- Parsing
- Compresión

## 12.- BIBLIOGRAFÍA:

Alfred V. Aho, J.E. Hopcroft, Jeffrey D. Ullman.

Data Structures and Algorithms,

Addison-Wesley Series in Computer Science and Information Processing, 1983.

Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford

Introduction to Algorithms, 3ra edición.

MIT Press, 2009

Skiena, Steve.

The Algorithm Design Manual. 2da Edición.

Springer, 2008.

Steven S. Skiena y Miguel Revilla.

Programming Challenges

Springer Verlag, 2003.

Peter Brass.

Advanced Data Structures

Cambridge University Press, 2008

Joseph O'Rourke

Computational Geometry in C, 2nd Edition,

Cambridge University Press, 1998.

Mark de Berg, Otfried Cheong, Marc van Kreveld, Mark Overmars.

Computational Geometry: Algorithms and Applications, 3ra Edición.

Springer Verlag, 2008

Jianer Chen

Computational Geometry: Methods and Applications

Texas University, 1996.

Dan Gusfield

Algorithms On Strings, Trees, and Sequences,

Cambridge University Press, 2007.

J.E. Hopcroft, J.D. Ullman.

Introduction to Automata Theory, Languages, and Computation.

Addison-Wesley, 1979

Ravindra K. Ahuja, Thomas L. Magnanti, and James Orlin

Network Flows: Theory, Algorithms, and Applications

Prentice Hall, 1993

R.E.Tarjan

Data Structures and Network Algorithms

Society for Industrial and Applied Mathematics, Philadelphia USA, 1983

Ge Nong, Sen Zhang, Wai Hong Chan

Two Efficient Algorithms for Linear Time Suffix Array Construction. IEEE Trans. Computers 60(10): 1471-1484 (2011)

Juha Kärkkäinen, Peter Sanders, Stefan Burkhardt

Linear work suffix array construction. J. ACM 53(6): 918-936 (2006)

Yefim Dinitz

Dinitz' Algorithm: The Original Version and Even's Version. Essays in Memory of Shimon Even 2006: 218-240

1-c-

Actividades prácticas propuestas (puede adjuntarse en hojas separadas):

Esta materia tiene dos objetivos centrales. Por un lado, poner en práctica los conocimientos teóricos que se adquieren en los cursos de grado de la Lic. en Ciencias de la Computación (FCEN, UBA), implementando varios algoritmos que se usan para modelar y resolver diferentes situaciones problemáticas en forma eficiente. Por el otro, profundizar en el desarrollo e implementación de algoritmos y estructuras de datos avanzadas que no se ven en la carrera (e.g., pattern matching, geometría computacional, suffix array), teniendo en cuenta sus aspectos teóricos. La materia tiene un claro enfoque práctico y de laboratorio, en donde todos los algoritmos vistos se implementan para resolver distintos problemas.

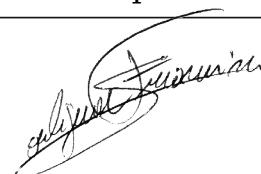
En función de estos objetivos, en cada clase se exhibirán diferentes soluciones a situaciones problemáticas, teniendo en cuenta los aspectos teóricos salientes de las mismas. Estas soluciones deben ser implementadas en código por los estudiantes a fin de resolver situaciones similares, justificando la corrección teórica de sus soluciones y la eficiencia algorítmica de las mismas.

(\*) Todos los cursos tendrán una validez de 5 años

(\*)(\*) Las actualizaciones de los docentes colaboradores son informados por la Dirección departamental al inicio de cada dictado del curso

Firma Subcomisión  
Doctorado

Firma del docente  
responsable



E-mail y teléfono del docente responsable

[fsouign@dc.uba.ar](mailto:fsouign@dc.uba.ar), 45763390 (int. a definir)

**Formulario para la presentación de Cursos de Posgrado/Doctorado - Res.  
CD2819/18 - ANEXO 2**

**Solicitud de Financiación**

**Año de presentación (\*)**

--

---

<b>Departamento docente que inicia el tramite:</b>

<b>Nombre del curso:</b>

<b>Nombre y Título del docente responsable:</b>

<b>Costo propuesto del curso por alumno (*):</b>

<b>Justificación del monto propuesto:</b>

(\*) Las excepciones aplicables para cada alumno serán consistentes con la reglamentación del Consejo Directivo que regula los aranceles y excepciones (Res. CD 484/13). El docente responsable del curso solicitará las excepciones por nota al consejo directivo a través de Mesa de Entradas.