



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Propuesta de Nuevo Plan de Estudios para la Licenciatura en Ciencias de la Computación

Preparada por la Comisión para la evaluación del cambio del Plan de Estudios designada por
CoDep

Buenos Aires, Diciembre de 2019

Introducción

Objetivo de este documento

Este documento presenta una propuesta de cambio del Plan de Estudios de la Licenciatura en Ciencias de la Computación dictada por el Departamento de Computación de la Facultad de Ciencias Exactas y Naturales de la UBA. La propuesta incluye además los cambios que aplican al título intermedio de Analista Universitario en Computación.

Autores de este documento y reconocimientos

La propuesta fue preparada por la Comisión para la Renovación del Plan de Estudios formada por:

- Pablo Brusco (G)
- Santiago Ceria (P)
- Tobías Carreira Munich (E)
- Gabriela Di Piazza (G)
- Agustín Gravano (P), hasta Octubre de 2019, luego Francisco Soullignac y Hernán Melgratti
- Gonzalo Guillamon (E)
- Christian Roldán (G)
- Uriel Rozenberg (E)

La comisión recibió además diversos aportes de docentes, graduados y alumnos del Departamento para mejorar las versiones iniciales. Queremos agradecer a todas estas personas por haberse tomado el trabajo de ayudarnos a elaborar esta propuesta en un marco de colaboración y trabajo en equipo.

Motivación

La conveniencia de revisar el Plan de Estudios está dada por los siguientes puntos:

- El plan vigente data del año 1993, es decir que fue preparado hace más de 26 años. Durante todo este tiempo la computación como disciplina ha tenido muchos avances, que se aceleraron de manera drástica en los últimos 10. Sólo por mencionar un ejemplo, la Comisión Curricular de la ACM / IEEE ha liberado varias versiones de recomendaciones de cambios durante estos años.
- Más allá de que haya sido muy exitoso, el plan tiene algunos problemas de diseño. El plan actual fue “revisado” informalmente a partir de cambios en el contenido de las materias que fueron realizando los distintos planteles docentes, pero esos cambios en muchos casos no fueron acordados o sincronizados con otras materias. De esta forma, el plan fue perdiendo integridad conceptual.

- Hay varios puntos del plan vigente que fueron sujeto de debate durante todos estos años, como la obligatoriedad y formato de la Tesis de Licenciatura, pero nunca fueron abordados a partir de un proceso de revisión con participación amplia de los tres claustros.
- Creemos que resulta necesario hacer una revisión crítica del plan teniendo en cuenta la baja tasa de graduación que la carrera está teniendo en los últimos años (alrededor del 15% de los alumnos que ingresan a la Facultad luego de aprobar el CBC).
- A lo largo de estos últimos años se realizaron distintos esfuerzos para revisar el plan de estudios, incluyendo la creación de una comisión que preparó un informe en 2017 en el que hubo consenso en la conveniencia de realizar esta revisión.

Diseño del nuevo plan

Objetivos de la carrera

El objetivo de esta carrera es formar graduados/as con sólidos conocimientos sobre los fundamentos teóricos de la computación y su aplicación práctica para resolver una amplia gama de problemas vinculados con la disciplina.

La idea es que los graduados/as de esta carrera tengan la formación que les permita tanto dedicarse a la investigación científica para continuar trabajando en ampliar el cuerpo de conocimientos de la disciplina como a la aplicación práctica de los conocimientos adquiridos principalmente en el desarrollo de sistemas de software aplicables a distintos tipos de dominios, utilizando distintos tipos de tecnologías. Nuestros graduados/as deberán poder adaptarse fácilmente a los rápidos cambios de la tecnología por la solidez de sus conocimientos sobre los fundamentos de la computación.

Titulación

Denominación de la carrera:

Licenciatura en Ciencias de la Computación

Denominación del Título:

Licenciado/a en Ciencias de la Computación

Denominación del Título Intermedio:

Analista Universitario/a en Computación

Perfil del Graduado/a en Ciencias de la Computación

Alcances del Título

Las competencias profesionales específicas para un Licenciado en Ciencias de la Computación incluyen:

- Aplicar de manera amplia los conceptos de pensamiento computacional a la resolución de problemas
- Participar en distintas etapas del desarrollo de sistemas de hardware / software / telecomunicaciones para distintos dominios, particularmente en la especificación, el diseño, la programación y la verificación y validación de sus diferentes módulos.
- Evaluar, entender y mejorar implementaciones existentes de sistemas de hardware / software / telecomunicaciones.
- Iniciar y liderar técnicamente emprendimientos vinculados con aplicaciones de la computación
- Efectuar tareas de auditoría y peritajes sobre sistemas informáticos.
- Trabajar en proyectos de investigación para ampliar la frontera del conocimiento en las ciencias de la computación
- Ejercer la docencia universitaria sobre temas vinculados con la computación

Requisitos de Ingreso

Para ingresar a la carrera el estudiante deberá contar con el nivel secundario completo o bien cumplir con las condiciones de ingreso establecidas en la Resolución CS N° 6716/97 para personas mayores de 25 años que no hayan completado estudios de nivel secundario.

Estructura de la carrera de Licenciatura en Ciencias de la Computación

La carrera consta de dos ciclos de formación: el Primer Ciclo de Grado (Ciclo Básico Común) y el Segundo Ciclo de Grado.

Primer Ciclo de Grado: Ciclo Básico Común (CBC). Duración teórica: 1 (UN) año. Consta de 6 (SEIS) asignaturas.

Segundo Ciclo de Grado: Duración teórica de 8 (OCHO) cuatrimestres, con 17 (DIECISIETE) asignaturas obligatorias, además de una tesis final de Licenciatura junto con 16 créditos de materias optativas.

Todas las asignaturas obligatorias son de modalidad presencial.

Duración y Cantidad de Materias

Con este plan la carrera tendrá una duración total de 5 años, incluyendo el Ciclo Básico Común. Los primeros dos cuatrimestres del Segundo Ciclo de Grado requieren cursar dos materias cada uno, mientras que a partir del tercer cuatrimestre se deben cursar tres materias por cuatrimestre para poder completar la carrera en el tiempo previsto por el plan.

La carga horaria prevista para este plan de estudios es de 25 horas semanales de cursada, incluyendo clases teóricas, clases prácticas, taller y consultas, salvo durante el primer cuatrimestre que la carga horaria será de 20 horas. A partir del segundo cuatrimestre, cuando se cursen dos materias, una tendrá una dedicación de 15 horas semanales (3 veces por semana) y otra de 10 horas (dos veces por semana). Cuando se cursen 3 materias por cuatrimestre, dos tendrán dedicación de 10 horas (dos veces por semana) y una de 5 horas (una vez por semana). El Departamento deberá garantizar días y horarios para que los alumnos puedan cursar este plan como está previsto sin superposición de materias en los mismos horarios. En el primer cuatrimestre, que tiene una carga horaria de 20 horas, se busca dejar tiempo libre a los alumnos para que dispongan de más tiempo para dedicar al estudio y además puedan participar de talleres sobre tecnologías coordinados por el Departamento de Computación que faciliten su inserción al mundo de la informática y la realización de trabajos prácticos (entornos de desarrollo, sistemas operativos, sistemas de control de versiones, etc.)

Trabajo Final - Tesis de Licenciatura

Al diseñar este plan de estudios se consideró conveniente que los alumnos pasen por una experiencia académica que vaya más allá de cursar materias y aprobar exámenes, manteniendo la idea de que la carrera tenga una tesis de licenciatura. Por otro lado, se buscó estructurar esta tesis de tal forma que no se transforme en un obstáculo para la graduación de los alumnos debido a su prolongación en el tiempo o la dificultad para cumplimentar sus distintas etapas. Para lograr estos objetivos se reservará un cuatrimestre con una dedicación de 15 horas semanales para que los alumnos puedan realizar la tesis en el marco de una materia nueva llamada "Trabajo final - Tesis de Licenciatura". En los siguientes puntos se resumen las características de esta materia que tiene como resultado final la aprobación de la tesis:

- Si bien la Tesis de Licenciatura no tendrá materias correlativas, los alumnos sólo podrán inscribirse y comenzar a cursarla cuando cuenten con tres o menos finales pendientes de materias obligatorias y hayan aprobado todas las cursadas de esas materias.
- Los alumnos podrán realizar una tesis del tipo "Trabajo Profesional" de manera individual o en grupos reducidos de hasta tres integrantes. Este tipo de tesis consistirán de un trabajo integrador de conocimientos adquiridos en distintas materias que normalmente será un desarrollo de software en un entorno real o simulado cubriendo varias etapas del ciclo de vida de un sistema de software. Será responsabilidad de los alumnos de la materia proponer el desarrollo a realizar junto con sus objetivos y

resultados esperados. El plantel docente de la materia irá haciendo el seguimiento del proyecto y ayudará a los equipos a lograr los objetivos planteados. La aprobación de este trabajo junto con su informe final correrá por cuenta del o los profesores a cargo de la materia Trabajo Final / Tesis de Licenciatura.

- Si los alumnos desean orientar su tesis hacia un tema académico, podrán trabajar de manera individual o en grupos de a lo sumo dos personas dentro del marco de esta misma materia. Estos trabajos finales habitualmente implican el estudio de un conjunto de papers, la elaboración de un reporte con conclusiones, y la reproducción de algún resultado de la literatura incluyendo algún desarrollo de software que puede incluir una cuota de originalidad. Es esperable que estos trabajos sean la base para la elaboración de un artículo técnico que pueda ser presentado a una conferencia nacional.
- Los alumnos deberán aprobar el trabajo final, sea del tipo profesional o académico, a lo sumo en las fechas de final del cuatrimestre siguiente al de la cursada. Si esto no ocurre, se deberá recursar la materia y realizar un trabajo nuevo.
- La materia se podrá aprobar además de forma “libre” mediante la presentación de un informe final y la defensa ante un jurado de un trabajo avalado por un profesor externo a la materia.

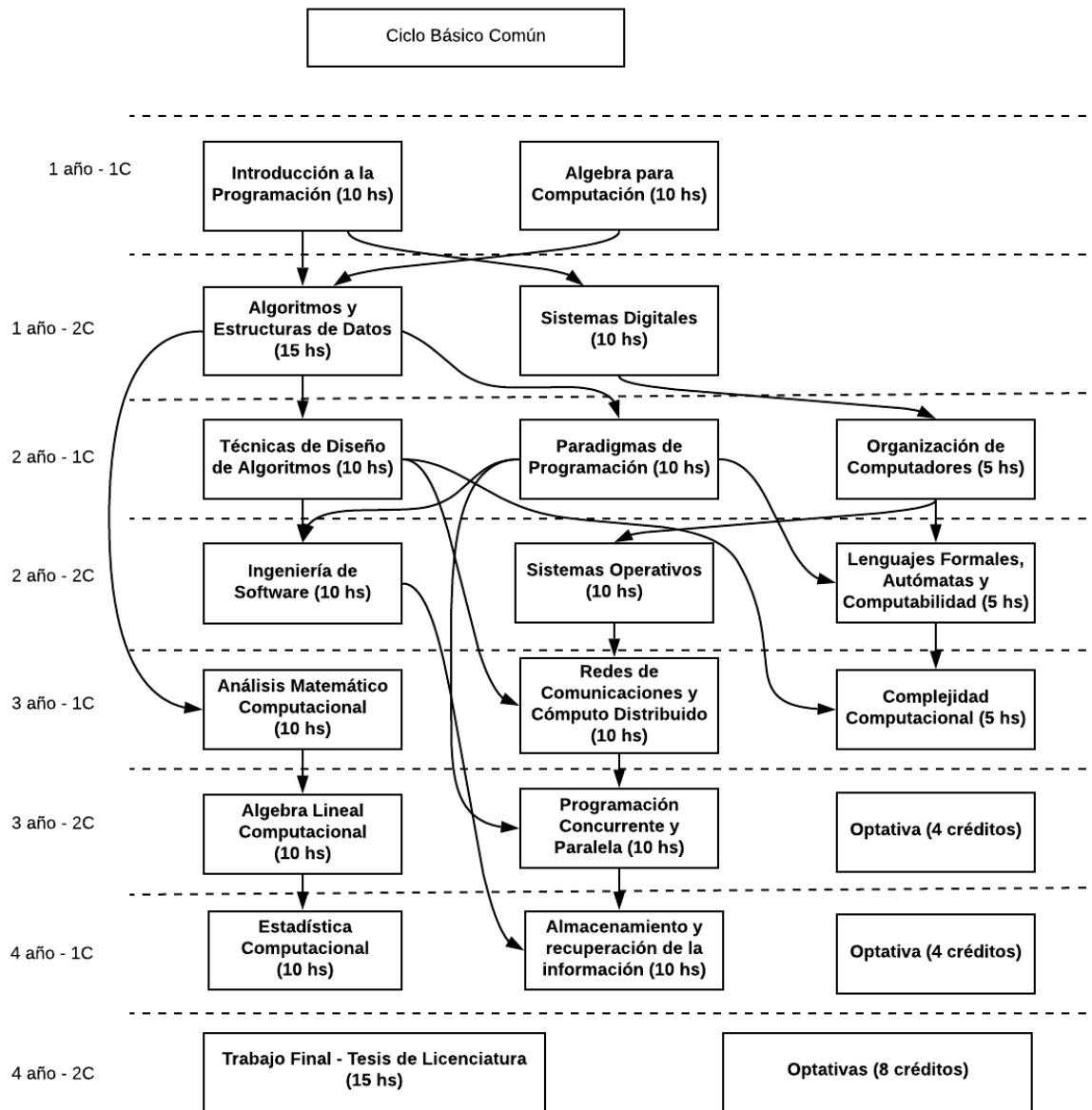
Título intermedio

Se otorgará el título intermedio de Analista Universitario en Computación al aprobar todas las materias hasta el segundo año completo y la materia Redes de Comunicaciones y Cómputo Distribuido de tercer año.

Plan de estudios propuesto

Materias y Correlatividades

El siguiente gráfico muestra las materias del plan en la trayectoria esperada. Las materias están unidas por flechas que representan correlatividades. Nótese en el gráfico que la tercera columna representa a las materias de 5 horas de dedicación semanal, que se cursan una vez por semana. El alumno tendrá la opción de adelantar la cursada de materias optativas, dejando para más adelante otras materias que no tienen correlatividades que las precedan.



Descripción de alto nivel de las materias

Materia	Objetivo	Correlativas	Dedicación
Introducción a la Programación (IP).	Introducir a los alumnos en el pensamiento algorítmico. Describir conceptos y estructuras algorítmicas fundamentales a través del estudio de la programación imperativa y funcional.	CBC	10 hs / sem
Álgebra para Computación (AC).	Cubrir los temas de matemática discreta que forman la base sobre los que se sustentan distintos temas de las ciencias de la computación.	CBC	10 hs / sem

Algoritmos y Estructuras de Datos (AED)	Utilizar técnicas de especificación, validación y verificación formal de algoritmos. Definir los tipos abstractos de datos fundamentales, diseñando estructuras de datos y algoritmos eficientes para su implementación, entendiendo cómo impactan las distintas decisiones estructurales en la complejidad de los algoritmos y en la interfaz de los tipos abstractos."	IP, AC	15 hs / sem
Sistemas Digitales (SD)	Presentar los temas fundamentales vinculados con los dispositivos de cómputo, incluyendo sistemas de representación, lógica digital y estructura de computadoras.	IP	10 hs / sem
Técnicas de Diseño de Algoritmos (TDA)	Describir las principales técnicas usadas para diseñar algoritmos a partir de la identificación de características comunes en los problemas, como por ejemplo búsqueda iterativa, divide and conquer, backtracking, etc, aplicando las distintas técnicas a problemas clásicos de grafos, cadenas y problemas geométricos.	AED	10 hs / sem
Paradigmas de Programación (PP)	Describir distintos paradigmas de programación con su aplicabilidad, ventajas y desventajas, preparando a los alumnos para que puedan entender, comparar y aplicar distintos tipos de lenguajes.	AED	10 hs / sem
Organización de Computadores (OC)	Describir la estructura básica computadores de propósito general y dedicado.	SD	5 hs / sem
Ingeniería de Software (IS)	Brindar los conocimientos básicos sobre cómo construir software en escala, sobre todo en lo referido a especificaciones, diseño y verificación / testing.	TDA, PP	10 hs / sem
Sistemas Operativos (SO)	Describir la arquitectura de un sistema operativo moderno y el funcionamiento de las distintas componentes de tal forma que los alumnos tengan los elementos básicos para su construcción.	OC	10 hs / sem
Análisis Matemático Computacional (AMC)	Proveer elementos de análisis matemático multivariado junto a métodos numéricos asociados para distintas aplicaciones en el ámbito de la computación y en particular vinculados con estadística computacional y el modelado/simulación.	AED	10 hs / sem
Álgebra Lineal computacional (ALC)	Describir las principales herramientas del álgebra lineal con una perspectiva computacional.	AMC	10 hs / sem
Redes de Comunicaciones y Cómputo Distribuido (RCD)	Brindar a los alumnos fundamentos teóricos y consideraciones prácticas sobre cómo funcionan las redes de comunicaciones y los sistemas de cómputo distribuido modernos.	SO, TDA	10 hs / sem
Lenguajes formales, Autómatas y Computabilidad (LAC)	Introducir a los alumnos en las estructuras de autómatas y los lenguajes que estas estructuras pueden definir, junto con aspectos de computabilidad de problemas.	PP, OC	5 hs / sem
Estadística Computacional (EC)	Brindar una introducción a los temas de probabilidad y estadística, con foco en inferencia estadística y aprendizaje automático.	ALC	10 hs / sem

Programación Concurrente y Paralela (PCP)	Describir los principales tipos de algoritmos y estructuras de datos concurrentes junto con sus limitaciones y posibles soluciones, incluyendo el aprovechamiento de procesamiento paralelo.	RCD	10 hs / sem
Complejidad Computacional (CC)	Definir el concepto de clases de complejidad de problemas computacionales, entendiendo cómo se clasifican los distintos problemas de acuerdo a la cantidad de recursos que consumen.	TDA, LFA	5 hs / sem
Almacenamiento y Recuperación de Información (ARI)	Cubrir el diseño y funcionamiento de sistemas de almacenamiento y recuperación de información, incluyendo distintos tipos de bases de datos y tópicos de gobierno de datos.	IS, PCP	10 hs / sem

Seminario sobre Tecnología y Sociedad

Como requisito para la graduación, tanto del título intermedio como de la carrera de grado, los alumnos deberán aprobar de manera adicional un Seminario de Tecnología y Sociedad que cubrirá temas de profesionalismo y ética profesional y el impacto en la sociedad del desarrollo y uso de tecnologías vinculadas con la computación, incluyendo el Pensamiento Latinoamericano sobre estos temas. Este seminario tendrá una duración aproximada de 2 meses cursando una vez por semana y se dictará todos los cuatrimestres.

Talleres para alumnos de primer año

Si bien no forman parte formal del plan, desde esta Comisión se invita al Departamento de Computación a que busque la forma de dictar talleres que faciliten la inserción de los alumnos en la carrera desde el punto de vista de las tecnologías a usar (Linux, sistemas de control de versiones, entornos de desarrollo, sistemas de composición de texto, etc). Estos talleres deberían dictarse idealmente en el día que los alumnos no cursen una materia.

Materias Optativas que forman parte del Plan

Las siguientes materias optativas deberán ser consideradas prioritarias por el Departamento y dictadas al menos una vez cada dos años.

- Seguridad Informática, con foco en el desarrollo de software seguro y conceptos de detección de vulnerabilidades
- Inteligencia Artificial, brindando un panorama amplio de las distintas ramas de esta temática, incluyendo tanto los enfoques simbólicos como los distintos paradigmas del aprendizaje automático
- Optimización, con foco en programación lineal entera y heurísticas asociadas
- Gestión de Proyectos de Software, brindando un panorama sobre las distintas formas de organizar esfuerzos de desarrollo de software.
- Métodos Formales de la Ingeniería de Software, con foco en especificaciones y “Model Checking”.

- Computación Cuántica.

Se espera que esta lista sea revisada y aprobada por el Consejo Departamental del Departamento de Computación cada año.

Temas propuestos para las materias

A continuación se presentan los temas propuestos para las materias. La segunda columna contiene los temas que deben ser considerados formalmente como parte del plan de estudios. La tercera columna incluye los temas que, al momento de preparar el plan de estudios, los miembros de la comisión y especialistas consultados consideran que pueden cubrirse cuando se implemente el plan. Se entiende que a medida que pasen los años la lista detallada de temas de la columna 3 puede variar, mientras que los temas de alto nivel de la columna 2 deben mantenerse. Por otro lado, también es esperable que la lista de temas de la columna 2 y la columna 3 sean muy similares o iguales para aquellas materias donde hay mayor estabilidad en el cuerpo de conocimientos, como por ejemplo las materias de Matemáticas. La implementación o no de los “temas sugeridos” quedará a criterio de los docentes que finalmente sean los responsables de implementar las materias una vez que el plan se ponga en marcha.

Materia	Temas del Plan	Temas sugeridos 2019
Introducción a la Programación (IP).	Algoritmos y resolución de problemas. Introducción al desarrollo de software mediante programación imperativa y funcional y sus principales estructuras y tipos de datos. Introducción al cálculo de complejidad algorítmica.	Panorama general de la computación, incluyendo descripción de la carrera. Algoritmos y resolución de problemas. Variables, expresiones y tipos. Herramientas de abstracción y modularización. Separación de comportamiento e implementación. Noción de contrato (informal). Estructuras de control: secuencia, selección, iteración y recursión. Tipos de datos: booleanos, enteros, tipos enumerados, tuplas, registros (structs), listas. Tipos abstractos de datos: interfaz pública vs. representación privada. Arreglos, pilas, colas, conjuntos, diccionarios. Nociones básicas de modularización (ej. clases) . Nociones básicas de testing. Cálculo de operaciones elementales, análisis de peor caso de una función
Álgebra para Computación (AC).	Lógica proposicional y de primer orden: sintaxis y semántica formal. Niveles de verdad (fórmulas satisfacibles, válidas y universalmente válidas). Valuaciones y estructura para primer orden. Noción De Consecuencia Semántica. Caracterización de clases de estructuras. Teoría de números. Teoría de conjuntos, relaciones. Estrategias de demostración. Inducción y recursión. Probabilidad discreta. Nociones básicas de teoría de grafos.	Lógica proposicional: sintaxis, semántica, tautología, consecuencia semántica, conjunto satisfacible. Lógica de primer orden: términos, fórmulas, variables libres y ligadas, interpretación, valuación, niveles de verdad, consecuencia semántica, conjunto satisfacible. Caracterización de estructuras: fórmulas que expresan propiedades. Teoría de conjuntos. Relaciones y funciones. Estrategias de demostración: demostración directa, modus ponens, razonamiento por el absurdo, demostración por la

		<p>contrarrecíproca, ejemplos y contraejemplos. Números naturales. Inducción en números naturales. Números enteros. Divisibilidad (algoritmo de Euclides, MCD, congruencia módulo, números primos). Nociones básicas de teoría de grafos. Tipos de datos inductivos. Nociones básicas de probabilidad discreta.</p>
Algoritmos y Estructuras de Datos (AED).	Nociones básicas de especificaciones formales de software, validación, verificación. Análisis y diseño de estructuras de datos fundamentales y sus principales algoritmos. Modularización.	<p>Especificación formal de software. Introducción a la validación y verificación. Métodos de demostración formal de correctitud (ej. weakest precondition). Teorema del invariante. Análisis básico de algoritmos: análisis asintótico, modelo RAM. Caso peor, promedio y mejor. Impacto de la complejidad algorítmica en el diseño de estructuras de datos. Manejo de memoria dinámica: punteros, listas enlazadas. Algoritmos de búsqueda y ordenamiento básicos y avanzados. String matching. Estructuras para búsqueda y ordenamiento: árboles de búsqueda, árboles balanceados, árboles digitales, hashing, colas de prioridad.</p>
Sistemas Digitales (SD)	Estructura de un computador. Sistemas de numeración y representación. Lógica digital. Diseño RTL. Circuitos secuenciales sincrónicos. Diseño lógico sobre dispositivos lógicos programables. Softcores.	<p>Estructura de un computador Sistemas de Numeración y representación. Bases de numeración, sistemas de codificación (ASCII UTF, etc) Lógica Digital: Compuertas, Diseño y optimización de funciones: Mapas K. Diseño RTL (Register Transfer Logic). Somera introducción a un HDL. Multiplexores / Demultiplexores. Símbolo tabla de verdad código HDL. Codificadores / Decodificadores. Símbolo tabla de verdad código HDL. Three state (High Z). Circuitos secuenciales sincrónicos. Latch D. Flip Flop D. Sumadores, Contadores. Diseño Lógico sobre Dispositivos Lógicos Programables. Estructura de una FPGA. LUTs (Look Up Tables). Formalización de un Lenguaje de Descripción de Hardware (Verilog o VHDL). Datapath. Unidad de control. Registros. Protocolos de Acceso a Buses. Softcores. CPU, Memoria, e I/O. Diseño de un juego simple de instrucciones: OPCODES. Modos de direccionamiento. Síntesis de una CPU sencilla programada con las instrucciones diseñadas</p>
Técnicas de Diseño de Algoritmos (TDA)	Conceptos básicos de técnicas algorítmicas y técnicas principales: búsqueda iterativa, divide and conquer, backtracking, programación dinámica y búsqueda local. Recorrido de grafos. Aplicación de las diferentes técnicas a problemas de grafos, cadenas y	<p>Conceptos básicos y utilidad de las técnicas algorítmicas. Divide and conquer: definición, cálculo de complejidad, teorema maestro y aplicación a problemas de ordenamiento, matrices y geometría (ej. puntos cercanos, cápsula convexa).</p>

	geometría. Resolución heurística de problemas de optimización.	Recorridos de grafos y árboles usando DFS y BFS; aplicación a problemas de grafos, incluyendo cálculos de distancias y cómputo de componentes. Búsqueda iterativa y backtracking: árbol de backtracking, implementación, tipos de recorridos y aplicación a problemas, incluyendo mochila y problemas de grafos (coloreo, clique máxima). Programación dinámica: principio de optimalidad, superposición de subproblemas, memorización; aplicación a problemas de strings y grafos, incluyendo camino mínimo. Algoritmos golosos; aplicación a problemas de grafos incluyendo árbol generador mínimo. Búsqueda local y su aplicación a problemas incluyendo flujo máximo en grafos. Algoritmos aproximados y heurísticas para problemas intratables, incluyendo clique, coloreo y viajante de comercio. Diseño de estructuras de datos como soporte para los distintos algoritmos (e.g., colas de prioridad, disjoint sets, listas de adyacencias, árboles sobre vectores, etc.)
Paradigmas de Programación (PP)	Características básicas de los distintos tipos de lenguajes de programación. Fundamentos de los principales paradigmas.	Sintaxis, Intérpretes y Compiladores de Lenguajes. Teoría de tipos. Metaprogramación. Fundamentos de los paradigmas funcional, imperativo, lógico y OO. Semántica operacional. Inferencia automática de reglas lógicas y árboles de decisión a partir de datos.
Organización de Computadores (OC)	Computadores de propósito dedicado vs. propósito general. Computadores dedicados: microcontroladores.	Computadores de Propósito dedicado vs propósito general. Organización general de un sistema de cómputo. Similitudes y diferencias, entre computadores dedicados y de propósito general. Aplicación de cada uno. Computadores dedicados: Microcontroladores. Estructura de la CPU. Memoria. Gestión de I/O (pooling vs interrupciones). Periféricos. Computadores de Propósito general. Sistema de memoria: Memoria Cache. Principio de vecindad o localidad. Cache asociativo. Protocolos de coherencia (MESI). Máquina de ejecución de instrucciones. Pipeline, superescalar, Predictores de salto, Ejecución fuera de orden. Descripción de una FPU.
Ingeniería de Software (IS)	El proceso de desarrollo de software. El software como modelo computable de la realidad. Diseño de sistemas. Técnicas modernas de construcción de software a mediana / gran escala.	Nociones básicas sobre el proceso de desarrollo de software. Especificaciones. Modelado del dominio del problema y modelado del dominio de la solución. Diseño Orientado a Objetos. Domain Driven Design. Patrones de diseño. Readaptabilidad. Introducción a la Validación y Verificación. Jerarquía de testing. Integración Continua. Test Driven Development. Técnicas de análisis de código.
Sistemas Operativos (SO)	Arranque de un sistema de cómputo. Hardware de soporte a un sistema operativo. Conceptos arquitecturales de sistemas operativos.	Arranque de un sistema de cómputo Hardware de soporte a un sistema operativo. Gestión de Memoria. Segmentación vs. paginación descripción de una MMU.

		<p>Memoria Virtual. Gestión de Interrupciones y excepciones. Sistema de Protección. Gestión de tareas.</p> <p>Conceptos arquitecturales de sistemas operativos. Scheduling. Alocación de memoria. Gestión de procesos.</p> <p>Intercomunicación. Device Drivers.</p> <p>Sincronización. Seguridad.</p>
Análisis Matemático Computacional (AMC)	<p>Funciones en varias variables. Cálculo diferencial. Extremos de funciones</p> <p>Aritmética del computador. Interpolación. Optimización no lineal.</p>	<p>Funciones en varias variables. Cálculo diferencial. Extremos de funciones</p> <p>Aritmética del computador. Análisis de errores numérico. Interpolación. Optimización no lineal. Sistemas de ec no lineales.</p> <p>Convergencia y orden de convergencia.</p>
Álgebra lineal computacional (ALC)	<p>Espacios vectoriales. Sistemas lineales. Aproximación.</p>	<p>Espacios vectoriales y bases. Productos internos y normas. Sistemas lineales.</p> <p>Autovalores. Métodos iterativos para sistemas lineales. Aproximación.</p>
Redes de Comunicaciones y Cómputo Distribuido (RCD)	<p>Teoría de la información. Principales arquitecturas, capas y protocolos de red.</p> <p>Nociones básicas sobre plataformas de cómputo distribuido.</p>	<p>Introducción a las redes fijas y móviles. Teoría de la información. Principales arquitecturas, capas y protocolos de red. Análisis de desempeño, escalabilidad, seguridad y confiabilidad. Plataformas de cómputo distribuido tipo cloud, fog y edge públicas y privadas. Orquestación de servicios, virtualización de redes y servicios. Cómputo y red definidos por software. Internet de las cosas. Blockchain.</p>
Lenguajes formales, Autómatas y Computabilidad (LAC).	<p>Lenguajes y su clasificación. Autómatas para decidir y enumerar los distintos tipos de lenguajes. Lenguajes no decidibles y no enumerables. Determinismo y no determinismo. Concepto general de algoritmo (Tesis de Church-Turing).</p>	<p>Definición de lenguaje y jerarquía de Chomsky.</p> <p>Lenguajes regulares: gramática, autómatas finitos determinísticos y no determinísticos, expresiones regulares. Lenguajes no regulares y lema de pumping.</p> <p>Lenguajes libres de contexto: gramática, autómatas de pila, lenguajes determinísticos. Gramáticas LR y nociones básicas de su aplicación a problemas de parsing. Lenguajes no libres de contexto y lema de pumping.</p> <p>Máquinas de Turing (determinísticas) y función parcial computable. Tesis de Church. Intérprete universal. Teorema del parámetro, recursión y punto fijo. Halting problem. Diagonalización. Lenguajes computables y computablemente enumerables. Teorema de Rice.</p>
Estadística Computacional (EC)	<p>Principios y aplicaciones de la estadística para la resolución con la computadora de problemas basados en datos tales como validación de hipótesis, exploración, clasificación y regresión Inferencia estadística.</p>	<p>VARIABLES aleatorias discretas y continuas. Probabilidad. Distribuciones y técnicas de muestreo.</p> <p>Estadística descriptiva. Test de hipótesis. Inferencia estadística.</p> <p>Algoritmos de regresión y clasificación. Ajuste y sobreajuste. Regularización. Técnicas y métricas de evaluación y selección de modelos (validación cruzada, precisión, recall, etc.).</p>

<p>Programación Concurrente y Paralela (PCP)</p>	<p>Concurrencia y paralelismo. Conflictos de concurrencia y mecanismos de sincronización. Principales algoritmos distribuidos.</p>	<p>Concurrencia vs. paralelismo. Modelos de concurrencia: memoria compartida, modelo de actores, pasaje de mensajes . Programación concurrente. Propiedades y conflictos de concurrencia: safety y liveness; deadlocks, livelocks, starvation, race conditions. Mecanismos de sincronización: semáforos, mutexes, monitores, barreras . Nociones básicas de paralelismo. Análisis de algoritmos paralelos. Complejidad (trabajo y span). Estructuras de datos inmutables. Algoritmos distribuidos: leader election, atomic commit, exclusión mutua, búsquedas distribuidas, algoritmos de consenso.</p> <p>Problemas clásicos: Acuerdo bizantino, Intuición de safety, liveness, fairness, Algoritmo del banquero, Panadería de Lamport, Exclusión mutua y locks distribuidos, Elección de líder.</p>
<p>Complejidad Computacional (CC)</p>	<p>Concepto de reducción de lenguajes. Concepto de complejidad temporal y espacial y clases de complejidad de lenguajes. Noción de problemas tratables e intratables. Complejidad basada en circuitos.</p>	<p>Equivalencia entre lenguaje y problema de decisión. Concepto de reducción entre lenguajes: reducciones many-one y pruebas de no-computabilidad de conjuntos. Conjuntos Σ_n y Σ_{n-1}-completos. Máquinas de Turing no determinísticas. Complejidad temporal: complejidad en distintos modelos de cómputo, equivalencia polinomial de modelos de cómputo. Determinismo versus no determinismo. P vs. NP: reducciones de tiempo polinomial. Clase NP-completo. Teorema de Cook. Otras clases de complejidad temporal y sus reducciones (ej. EXPTIME, etc). Complejidad espacial: PSPACE vs. NPSPACE, teorema de Savitch. Clase LOGSPACE, algoritmos LOGSPACE y reducciones LOGSPACE. Clases de complejidad espacial y su relación con clases temporales. Problemas PSPACE-completos. Tratabilidad: explicación del teorema de la jerarquía temporal y sus consecuencias. Circuitos booleanos, complejidad basada en circuitos y su relación con complejidad sobre MTs.</p>
<p>Almacenamiento y Recuperación de Información (ARI)</p>	<p>Modelado de datos. Principales modelos para almacenamiento y recuperación de datos. Tópicos de gobierno de datos.</p>	<p>Modelado de datos. Lenguajes de manipulación de datos. Pautas de diseño de bases de datos. Transacciones. Limitaciones del modelo relacional y modelos alternativos de bases de datos. Optimización de Consultas. Recuperación y Control de Concurrencia. Bases de Datos distribuidas y paralelas. Extracción de Información. Tópicos de gobierno de datos. Pautas de seguridad y normas legales vinculadas a los datos.</p>

Dedicación horaria total

El siguiente cuadro detalla la dedicación horaria total de la carrera, incluyendo el Ciclo Básico Común, y su comparación con los mínimos requeridos para Licenciaturas por parte del Ministerio de Educación. La tabla muestra que la carrera tiene una dedicación horaria total de 3808 horas, es decir 608 horas más que lo requerido por el Ministerio. Sólo a efectos comparativos, la Licenciatura en Análisis de Sistemas de la Facultad de Ingeniería de la UBA, cuyo plan fue aprobado en el año 2014, tiene una carga horaria total de 3424 horas.

CBC	Hs x Sem	Total hs
Introducción al Conocimiento Científico	4	64
Introducción al Conocimiento de la Sociedad y el Estado	4	64
Análisis Matemático	9	144
Álgebra	9	144
Física	6	96
Química	6	96
Total CBC		608
Introducción a la Programación	10	160
Álgebra para Computación	10	160
Algoritmos y Estructuras de Datos	15	240
Sistemas Digitales	10	160
Técnicas de Diseño de Algoritmos	10	160
Paradigmas de Programación	10	160
Organización de Computadores	5	80
Ingeniería de Software	10	160
Sistemas Operativos	10	160
Análisis Matemático Computacional	10	160
Álgebra Lineal Computacional	10	160
Redes de Comunicaciones y Cómputo Distribuido	10	160
Lenguajes Formales, Autómatas y Computabilidad	5	80
Estadística Computacional	10	160
Programación Concurrente y Paralela	10	160
Complejidad Computacional	5	80
Almacenamiento y Recuperación de la Información	10	160

Total Obligatorias		2560
Trabajo Final / Tesis de Licenciatura	15	240
Optativas	20	320
Total General (CBC, Obligatorias, Tesis, Optativas)		3728
Mínimo requerido por Ministerio de Educación		3200
Diferencia (a favor)		528

Comparación de la Dedicación con el Plan Actual

El plan actual de la Licenciatura (1993) tiene la siguiente dedicación para las materias obligatorias, que en el plan nuevo suman 2560 horas:

Materia	Hs x sem	Total hs
Algebra 1	10	160
Análisis 2	10	160
Probabilidad y Estadística	10	160
Métodos Numéricos	9	144
Algoritmos y Estructuras de Datos 1	15	240
Algoritmos y Estructuras de Datos 2	15	240
Algoritmos y Estructuras de Datos 3	15	240
Paradigmas de Programación	7	112
Organización del Computador 1	15	240
Organización del Computador 2	10	160
Sistemas Operativos	9	144
Teoría de las Comunicaciones	9	144
Lógica y Computabilidad	7	112
Teoría de Lenguajes	9	144
Ingeniería de Software 1	9	144
Ingeniería de Software 2	9	144
Base de Datos	9	144
Total		2832

Esto significa que desde el punto de vista del plan diseñado en cuanto a materias obligatorias, el nuevo plan tiene 272 horas menos que el plan actual (aproximadamente un 10%).

Ciclo lectivo a partir del cual tendrá vigencia

El nuevo plan se implementará a partir del cuatrimestre posterior a su aprobación por parte del Consejo Superior de la UBA.

Equivalencias y Plan de Transición

El plan 1993 mantendrá su vigencia por 9 (nueve) cuatrimestres contados a partir de la fecha de entrada en vigencia del nuevo plan de estudios.

Los estudiantes que tengan aprobado el Primer Ciclo completo y por lo menos una asignatura del Segundo Ciclo con el plan 1993 podrán optar por continuar con dicho plan.

Los estudiantes que opten por el pase de plan, cambio que pueden hacer en cualquier momento, obtendrán créditos para el nuevo plan por las asignaturas del plan 1993 que hayan aprobado a la fecha de opción del pase de plan, según lo indicado en la tabla de equivalencias.

Adicionalmente, se les reconocerán créditos por el total de materias optativas del plan 1993 que hayan aprobado a la fecha de opción de pase de plan.

Los estudiantes del nuevo plan no podrán optar por pasarse al plan anterior.

Equivalencias con Materias de Otras Carreras

Las materias Álgebra para Computación y Análisis Matemático Computacional serán variaciones de las materias Álgebra I y Análisis II dictadas por el Departamento de Matemática y que son obligatorias para varias carreras de la Facultad. Estas dos últimas serán tomadas como válidas para los alumnos de la carrera de Ciencias de la Computación.

Tabla de equivalencias

Por haber aprobado la materia plan 1993	Se computará la materia del nuevo plan
Análisis Matemático II + Métodos Numéricos	Análisis Matemático Computacional
Algoritmos y Estructuras de Datos 1	Introducción a la Programación
Algebra 1	Álgebra para Computación
Organización del Computador 1	Sistemas Digitales, Organización de Computadores

Algoritmos y Estructuras de Datos 2	Algoritmos y Estructuras de Datos
Ingeniería de Software 1	Ingeniería de Software
Sistemas Operativos	Sistemas Operativos
Bases de Datos	Almacenamiento y Recuperación de la Información
Métodos Numéricos	Álgebra Lineal Computacional
Probabilidad y Estadística	Estadística Computacional
Lógica y Computabilidad + Teoría de Lenguajes	Lenguajes Formales, Autómatas y Computabilidad
Teoría de las Comunicaciones	Redes de Comunicaciones y Cómputo Distribuido
Algoritmos y Estructuras de Datos 3 + Lógica y Computabilidad	Complejidad Computacional + Técnicas de Diseño de Algoritmos
Paradigmas de Lenguajes de Programación	Paradigmas de Programación

El Consejo Departamental del DC podrá decidir computar como materias optativas la aprobación de otras materias obligatorias del plan 1993 que no tengan una materia equivalente en el nuevo plan.