



Universidad Nacional de Río Cuarto
Facultad de Ciencias Exactas, Físico-Químicas y Naturales

FORMULARIO PARA LA PRESENTACIÓN DE PROGRAMAS DE ASIGNATURAS en el CONTEXTO DE PANDEMIA por Covid-19¹

Año Lectivo: 2020

UNIVERSIDAD NACIONAL DE RÍO CUARTO FACULTAD DE CIENCIAS EXACTAS, FÍSICO-QUÍMICAS Y NATURALES DEPARTAMENTO DE COMPUTACIÓN

CARRERA/S: Analista en Computación y Profesorado en Ciencias de la Computación (Estructuras de Datos y Algoritmos), Licenciatura en Ciencias de la Computación (Algoritmos I)

PLAN DE ESTUDIOS: 1999

ASIGNATURA: Estructura de Datos y Algoritmos/Algoritmos I

CÓDIGO: 3325/3301

MODALIDAD DE CURSADO: Presencial

DOCENTE RESPONSABLE: Pablo Castro, Profesor Adjunto, Dedicación Simple.

EQUIPO DOCENTE: Lic. Sonia Permigiani, Prof. Ernesto Cerdá, Dr. Simon Gutierrez

RÉGIMEN DE LA ASIGNATURA: Cuatrimestral

UBICACIÓN EN EL PLAN DE ESTUDIO: 2do Año – 2do Cuatrimestre

RÉGIMEN DE CORRELATIVIDADES:

Asignaturas aprobadas: Introducción a la algorítmica y Programación

Asignaturas regulares: Programación Avanzada, Algebra (Algoritmos I)

CARÁCTER DE LA ASIGNATURA: Obligatoria

CARGA HORARIA TOTAL: 112 horas

Teóricas:	56hs	Prácticas:	56 hs	Teóricas - Prácticas: hs	Laboratorio:	56hs
------------------	-------------	-------------------	--------------	----------------------------------	----------------	---------------------	-------------

CARGA HORARIA SEMANAL: 12 horas (según el plan de estudio vigente)

Teóricas:	4 hs	Prácticas:	4 hs	Teóricas - Prácticas: hs	Laboratorio:	4 hs
------------------	-------------	-------------------	-------------	----------------------------------	----------------	---------------------	-------------

¹ Res. CS 120/2017 y Res. CD 049/2020



Universidad Nacional de Río Cuarto
Facultad de Ciencias Exactas, Físico-Químicas y Naturales

A. CONTEXTUALIZACIÓN DE LA ASIGNATURA

La asignatura está en el segundo año de las carreras citadas, el material abordado en la materia provee las herramientas básicas para poder abordar los cursos más avanzados de computación.

B. OBJETIVOS PROPUESTOS

El objetivo esencial de la materia es lograr que los alumnos se familiaricen con las técnicas más importantes de representación de datos, que comprendan los mecanismos de los lenguajes de programación modernos para su implementación, y sepan identificar los problemas prácticos en los cuales pueden emplearse cada una de las representaciones de datos estudiadas.

Un objetivo importante de la asignatura es que los alumnos se familiaricen con los conceptos fundamentales de la programación orientada a objetos, y realicen sus primeras actividades prácticas en relación al diseño y la implementación de programas orientados a objetos.

C. EJES TEMÁTICOS ESTRUCTURANTES DE LA ASIGNATURA Y ESPECIFICACIÓN DE CONTENIDOS

C.1. Contenidos mínimos (según plan de estudio vigente)

Introducción a la programación orientada a objetos. La noción de tipos abstractos de datos. Revisión de implementaciones clásicas de tipos abstractos de datos estructurados lineales. Listas, pilas y colas, y sus implementaciones clásicas. Medidas de tiempo de ejecución de programas. Análisis de tiempo de ejecución en caso promedio y peor caso. Análisis de tiempo de ejecución de algoritmos iterativos y recursivos. Tipos abstractos de datos más avanzados. El tipo abstracto de datos Grafo. Algoritmos de recorrido/visita de grafos: recorridos "primero en profundidad" y "primero a lo ancho". Implementaciones clásicas de grafos (matrices de adyacencias y listas de adyacencias). Grafos dirigidos y no dirigidos. Grafos con costos. Aplicaciones del tipo abstracto de datos Grafo. Algoritmos clásicos sobre grafos. Cálculo de camino de mínimo costo: Algoritmos de Dijkstra y Floyd. Clausura transitiva: Algoritmo de Warshall. El problema del árbol abarcador de costo mínimo: Algoritmos de Prim y Kruskal. Estructuras de datos avanzadas. Heaps y sus aplicaciones.

C.2. Ejes temáticos o unidades

Introducción a la programación orientada a objetos. Clases y objetos. Relaciones de agregación y uso. Herencia y polimorfismo. Genericidad. Introducción al diseño orientado a objetos. Conceptos de cohesión y acoplamiento. Principio abierto-cerrado.

La noción de tipos abstractos de datos. Representación de datos. Funciones de abstracción e invariantes de representación. Implementación de tipos abstractos de datos en lenguajes orientados a objetos. Especificaciones básicas y jerárquica. Corrección y completitud



Universidad Nacional de Río Cuarto
Facultad de Ciencias Exactas, Físico-Químicas y Naturales

Revisión de implementaciones clásicas de tipos abstractos de datos estructurados lineales. Listas, pilas y colas, y sus implementaciones clásicas. Comparación de las implementaciones clásicas de tipos abstractos de datos lineales.

Medidas de tiempo de ejecución de programas. Análisis de tiempo de ejecución en caso promedio y peor caso. Análisis de tiempo de ejecución de algoritmos iterativos y recursivos. Resolución de ecuaciones de recurrencias. Tasas de crecimiento de funciones de tiempo de ejecución. Análisis empírico del tiempo de ejecución de programas.

Tipos abstractos de datos más avanzados. Los tipos abstractos de datos Conjunto y Diccionario. Diccionarios y su implementación. Árboles como estructuras de datos. Recorridos sobre árboles (inorder, preorder, postorder). Árboles binarios de búsqueda. AVLs y árboles 2-3. Comparación de las implementaciones clásicas de diccionarios de acuerdo a la eficiencia de sus operaciones.

El tipo abstracto de datos Grafo. Algoritmos de recorrido/visita de grafos: recorridos "primero en profundidad" y "primero a lo ancho". Implementaciones clásicas de grafos (matrices de adyacencias y listas de adyacencias). Grafos dirigidos y no dirigidos. Grafos con costos. Aplicaciones del tipo abstracto de datos Grafo. Algoritmos clásicos sobre grafos. Cálculo de camino de mínimo costo: Algoritmos de Dijkstra y Floyd. Clausura transitiva: Algoritmo de Warshall. El problema del árbol abarcador de costo mínimo: Algoritmos de Prim y Kruskal.

Estructuras de datos avanzadas. Heaps y sus aplicaciones. Diagramas binarios de decisión (BDDs) y aplicaciones. Hashing.

D. ACTIVIDADES A DESARROLLAR

- E. Clases teóricas: 4 horas semanales de clases teóricas, divididas en dos clases de dos horas cada una; en estas clases se les enseña a los estudiantes la teoría básica de la materia.
- F. Clases práctica: 4 horas semanales de clases prácticas y de laboratorio, estas clases permiten a los estudiantes afianzar los conocimientos teóricos mediante la aplicación de la teoría a problemas medianamente complejos.
- G. Laboratorios: 4 horas semanales de laboratorios, en estas clases los estudiantes implementan en lenguajes de programación los conceptos vistos en los teóricos y prácticos.
- H. Horarios de Consulta: Cada docente provee horas de consulta adicionales para los prácticos, teóricos y laboratorios, estas son dos horas semanales adicionales.

D.1. Actividades en modalidad virtual (modalidades alternativas a la presencialidad).

Las clases se llevarán a cabo por la plataforma meet.

CLASES TEÓRICAS: Se llevarán a cabo en forma virtual, por google meet o SIAL, 4 horas por semana

CLASES PRÁCTICAS: Se llevarán a cabo en forma virtual, por google meet o SIAL, 4 horas por semana



Universidad Nacional de Río Cuarto
Facultad de Ciencias Exactas, Físico-Químicas y Naturales

CLASES DE TRABAJOS PRÁCTICOS DE LABORATORIO Se llevarán a cabo en forma virtual, por google meet o SIAL, también se utilizarán la plataforma GitHub, 4 horas por semana

OTRAS: instancias evaluativas, seminarios, talleres, coloquios, etc. (nómina, modalidad, metodología, recursos y carga horaria)

D.2. Actividades en la presencialidad

No se tienen planificadas actividades presenciales

I. PROGRAMAS Y/O PROYECTOS PEDAGÓGICOS INNOVADORES E INCLUSIVOS

J. CRONOGRAMA TENTATIVO DE CLASES E INSTANCIAS EVALUATIVAS a realizar en la virtualidad y en la presencialidad

Se tomará un parcial de forma virtual, con si respectivo recuperatorio. Además se tomarán trabajos prácticos que los alumnos deberán resolver y entregar por medio de la plataforma GitHub.

F.1. Cronograma tentativo de clases e instancias evaluativas a realizar en la virtualidad.

Semana	Día/Horas	Actividad: tipo y descripción*
1	26/08/20	Laboratorio
1	26/08/20	Laboratorio
1	28/08/20	Práctico
1	26/08/20	Teórico: Introducción a la materia
1	27/08/20	Práctico: Repaso
1	28/08/20	Teórico: Introducción a la materia
2	31/08/20	Laboratorio: Repaso
2	31/08/20	Laboratorio: Repaso
2	2/09/20	Práctico: Introducción a O.O
2	2/09/20	Teórico: Encapsulación
2	3/09/20	Práctico: Introducción a Git
2	4/09/20	Teórico: Programación Orientada a Objetos
3	7/09/20	Laboratorio: Introducción a los Compiladores a usar en la Materia
3	7/09/20	Laboratorio: Uso de los compiladores a usar en la Materia
3	9/09/20	Práctico: Ejercicios básicos de OO
3	9/09/20	Teórico: Introducción a Java
3	10/09/20	Práctico: Ejercicios en Java
3	11/09/20	Teórico: Introducción a Java
4	14/09/20	Laboratorio: Uso de Java



Universidad Nacional de Río Cuarto
Facultad de Ciencias Exactas, Físico-Químicas y Naturales

4	14/09/20	Laboratorio: Uso de Java
4	16/09/20	Práctico: Ejercicios en Java, estructuras de datos simples
4	16/09/20	Teórico: Excepciones en Java
4	17/09/20	Práctico: Ejercicios con Excepciones
4	18/09/20	Teórico: TAD y Algebras
5	21/09/20	Laboratorio: Problemas con Excepciones
5	21/09/20	Laboratorio: Implementaciones de TADS en JAVA
5	23/09/20	Práctico: Ejercicios básicos con TADS
5	23/09/20	Teórico: TAD y Algebras
5	24/09/20	Práctico: Ejercicios con algebras y TADS
5	25/09/20	Teórico: Funciones de Abstracción
6	28/09/20	Laboratorio: Implementaciones de Funciones de Abstracción
6	28/09/20	Laboratorio: Implementaciones de Funciones de Abstracción
6	30/09/20	Práctico: Ejercicios avanzados con TADs
6	30/09/20	Teórico: Tiempo de Ejecución
6	1/10/20	Práctico: Repaso de logaritmos, sumatorias , productorias etc
6	2/10/20	Teórico: Notación Big-Oh
6	5/10/20	Laboratorio: Implementación de ejercicios con análisis de tiempo de ejecución
6	5/10/20	Laboratorio: Implementación de ejercicios con análisis de tiempo de ejecución
6	7/10/20	Práctico: Ejercicios sobre Big-Oh
6	7/10/20	Teórico: Sorting I
6	8/10/20	Práctico: Resolución de problemas de Sorting
6	9/10/20	Teórico: Sorting II
7	12/10/20	Laboratorio: Implementación de Algoritmos de Sorting
7	12/10/20	Laboratorio: Implementación de Algoritmos de Sorting
7	14/10/20	Práctico: Analisis de tiempo de ejecución de sorting
7	14/10/20	Teórico: Árboles I
7	15/10/20	Práctico: resolución de problemas sobre árboles
7	16/10/20	Teórico; Árboles II
8	19/10/20	Laboratorio: Implementación de Árboles en JAVA-Rust
8	19/10/20	Laboratorio: Implementación de Árboles en JAVA-Rust
8	21/10/20	Práctico: Problemas adiciones sobre Árboles.
8	21/10/20	Teórico: Diccionarios
8	22/10/20	Práctico: Problemas básicos obre diccionarios
8	23/10/20	Teórico: Implementaciones Básicas de Diccionarios, ABB
9	26/10/20	Laboratorio: Implementación de ABBs en JAVA
9	26/10/20	Laboratorio: : Implementación de ABBs en JAVA
9	28/10/20	Práctico: Propiedades de ABBs
9	28/10/20	Teórico: AVL, Árboles 2-3
9	29/10/20	Práctico: Resolución de problemas sobre AVL y Árboles 2-3
9	30/10/20	Teórico: Colas de Prioridad - Heaps
10	2/11/20	Laboratorio: Implementación de AVL, Árboles 2-3
10	2/11/20	Laboratorio: Implementación de AVL, Árboles 2-3
10	4/11/20	Práctico: Propiedades de Heaps



Universidad Nacional de Río Cuarto
Facultad de Ciencias Exactas, Físico-Químicas y Naturales

10	4/11/20	Teórico: Funciones de Hashing
10	5/11/20	Práctico: Resolución de problemas sobre funciones de Hashing
10	6/11/20	Teórico: Grafos, Introducción
11	9/11/20	Laboratorio: Implementación de Grafos
11	9/11/20	Laboratorio: Implementación Grafos
11	11/11/20	Práctico: Resolución de Problemas sobre Grafos
11	11/11/20	Teórico: Implementaciones Básicas de Grafos
11	12/11/20	Práctico: Grafos en JAVA
11	13/11/20	Teórico: Algoritmos de recorridos de grafos
12	16/11/20	Laboratorio: Programación de Grafos en JAVA
12	16/11/20	Laboratorio: Programación de Grafos en JAVA
12	18/11/20	Práctico: Problemas sobre Grafos
12	18/11/20	Teórico: Algoritmos de Warshall y Algoritmo de Dijkstra
12	19/11/20	Práctico: Propiedades del algoritmo de Warshall y Dijkstra
12	20/11/20	Teórico: Spanning Trees, Algoritmo de Kruskal y Algoritmo de Prims
13	23/11/20	Laboratorio: Implementación de Warshall y Dijkstra
13	23/11/20	Laboratorio: Implementación de Warshall y Dijkstra
13	25/11/20	Práctico: Resolución de problemas sobre árboles cobertores
13	25/11/20	Teórico: Grafos Eulerianos, Aplicaciones de DFS
13	26/11/20	Práctico: Resolución de Problemas Usando DFS
13	27/11/20	Teórico: Introducción a Complejidad Computacional
14	30/11/20	Laboratorio: Implementación de Algoritmos sobre Grafos
14	30/11/20	Laboratorio: Implementación de Algoritmos sobre Grafos
14	2/12/20	Práctico: Ejercicios adicionales sobre grafos
14	2/12/20	Teórico: Problemas NP-Complejos
14	3/12/20	Práctico: Repaso
14	4/12/20	Teórico: Repaso de la Materia

*Teóricos, teóricos-prácticos, trabajos de laboratorios, seminarios, talleres, coloquios, instancias evaluativas, consultas grupales y/o individuales, otras.

F.2. Cronograma tentativo de clases e instancias evaluativas a realizar en la presencialidad.

Semana	Día/Horas	Actividad: tipo y descripción*

*Teóricos, teóricos-prácticos, trabajos de laboratorios, seminarios, talleres, coloquios, instancias evaluativas, consultas grupales y/o individuales, otras.

K. BIBLIOGRAFÍA



Universidad Nacional de Río Cuarto
Facultad de Ciencias Exactas, Físico-Químicas y Naturales

G.1. Bibliografía obligatoria y de consulta:

Mark Allen Weiss, " Data Structures and Algorithm Analysis in *Java*", Addison-Wesley, 2010
T.Cormen, C. Leiserson, R.Rivest. C.Stein, "Introduction to Algorithms", The MIT Press, 2001.
Claus Matzinger. Hands-On, Data Structures and Algorithms with Rust, Packt Publishing, 2019.

G.2. Plataformas/herramientas virtuales; materiales audiovisuales, otros.

Se utilizará Google Classroom como repositorio de materiales, apuntes, presentaciones, videos, etc. Las clases se llevarán a cabo por medio de Google Meet, y serán grabadas, Todo el materia estará disponibles por medio de las aulas virtuales del SIAL. También se utilizará la aplicación de mensajería instantánea Slack, y los repositorios para código de programación GitHub. Por el cual, se dejará disponible código de programación para que los alumnos puedan acceder al mismo, y modificarlo.

L. DÍA Y HORARIOS DE CLASES VIRTUALES y PRESENCIALES

Teóricos: Miercoles y Viernes de 14 a 16hs.

Practicos: Miercoles de 10 a 12hs, Jueves de 14 a 16hs.

Laboratorios: Lunes de 10 a 12hs y Lunes de 14 a 16hs

M. DÍA Y HORARIO DE CLASES DE CONSULTAS VIRTUALES y PRESENCIALES

Serán acordadas con los estudiantes.

N. REQUISITOS PARA OBTENER LA REGULARIDAD Y LA PROMOCIÓN

Para regularizar la materia se debe aprobar el parcial o su recuperatorio, además se deberán aprobar los trabajos prácticos dados durante la materia. La materia se puede rendir como libre para para ello se solicitará un proyecto de programación que deberá ser entregado antes del final.

O. CARACTERÍSTICAS, MODALIDAD Y CRITERIOS DE LAS INSTANCIAS EVALUATIVAS

El parcial se desarrollará de manera virtual por medio de las plataformas disponibles, de la misma forma se desarrollará el recuperatorio, además se darán trabajos prácticos que los estudiantes deberán entregar en los plazos establecidos.