

**Carrera: Ingeniería en Sistemas de Información****Asignatura:** Algoritmos y Estructuras de Datos**Planificación a partir del Ciclo Lectivo 2025****1. Datos administrativos de la asignatura**

Nivel en la carrera	1	Duración	Anual
Plan	2023		
Bloque curricular:	Tecnologías Básicas		
Carga horaria presencial semanal (hs. cátedra):	5	Carga Horaria total (hs. reloj):	120 horas reloj
Carga horaria no presencial semanal (hs. reloj) (si correspondiese)		% horas no presenciales (hs. reloj) (si correspondiese)	

**2. Presentación, Fundamentación**

La asignatura forma parte del Bloque de Tecnologías Básicas, de acuerdo a la categorización indicada por la Ordenanza 1877/22.

La asignatura AED cumple con los propósitos generales del área de Tecnologías Básicas desde la perspectiva del estudio y aplicación de técnicas y estrategias para el planteo de algoritmos y estructuras de datos y su implementación mediante lenguajes de programación para la resolución de problemas de ingeniería. Es la primera asignatura del Plan de Estudios orientada a la formación en técnicas de programación, y se convierte de esta forma en la asignatura base para el cursado de las que siguen en ciclos posteriores.

En relación al Perfil de Ingeniero (Ordenanza 1877/22) esta asignatura contribuye directamente a la formación analítica sólida, orientada a la interpretación y resolución de problemas mediante el empleo de metodologías y tecnologías de la información.

Y en relación a los Alcances del Título (Ordenanza 1877/22) esta asignatura contribuye en FORMACIÓN INICIAL Y BÁSICA para cualquiera de las actividades reservadas o para cualquiera de los alcances del el título que tengan relación con el desarrollo, puesta en operación, y control de sistemas de información en cualquier ámbito donde estos fuesen requeridos. Y como es lógico (al tratarse de una asignatura básica), la contribución esperada tanto al perfil como a los alcances es necesariamente de carácter introductorio.

**3. Relación de la asignatura con las competencias de egreso de la carrera**

En la tabla siguiente se establece la relación de la asignatura con las competencias de egreso: Específicas, Genéricas Tecnológicas y Genéricas Sociales, Políticas y Actitudinales de la carrera. Se incluyen las competencias de egreso a las que tributa, aportes reales y significativos de la asignatura, y en qué nivel (no aporta, bajo, medio, alto).

<b>Competencias</b>	<b>Nivel</b>
<b>Competencias genéricas tecnológicas (CG):</b>	
CG.1. Identificación, formulación y resolución de problemas de ingeniería en sistemas de información/informática.	Bajo
CG.2. Concepción, diseño y desarrollo de proyectos de Ingeniería en Sistemas de Información/Informática	No aporta
CG.3. Gestión, planificación, ejecución y control de proyectos de Ingeniería en Sistemas de Información/Informática.	No aporta
CG.4. Utilización de técnicas y herramientas de aplicación de Ingeniería en Sistemas de Información/Informática.	Bajo
CG.5. Generación de desarrollos tecnológicos y/o innovaciones tecnológicas.	Bajo
<b>Competencias genéricas sociales, políticas y actitudinales (CG)</b>	
CG.6. Fundamentos para el desempeño en equipos de trabajo.	Bajo
CG.7. Fundamentos para una comunicación efectiva.	Medio
CG.8. Fundamentos para una actuación profesional ética y responsable.	Medio
CG.9. Fundamentos para evaluar y actuar en relación con el impacto social de su actividad profesional en el contexto global y local.	No aporta
CG.10. Aprender en forma continua y autónoma.	Medio
CG.11. Fundamentos para el desarrollo de una actitud profesional emprendedora	No aporta
<b>Competencias Específicas de la carrera</b>	
CE1.1. Especificar, proyectar y desarrollar sistemas de información para concebir soluciones tecnológicas que permitan resolver situaciones en las organizaciones mediante el empleo de metodologías de sistemas y tecnologías asociadas a los sistemas de información.	No aporta
CE1.2. Especificar, proyectar y desarrollar sistemas de comunicación de datos, evaluando posibles soluciones tecnológicas disponibles para dar soporte a los sistemas de información en lo referido al procesamiento y comunicación de datos.	No aporta
CE1.3. Especificar, proyectar y desarrollar software para la elaboración de soluciones informáticas con el propósito de resolver problemas estratégicos y operativos, así como de servicios y de negocios, en el marco de una actividad económica que sea social y ambientalmente sustentable.	Bajo

CE2.1. Proyectar y dirigir lo referido a seguridad informática para seleccionar y aplicar técnicas, herramientas, métodos y normas, garantizando la seguridad y privacidad de la información procesada y generada por los sistemas de información.	No aporta
CE.3.1. Establecer métricas y normas de calidad de software para medir, evaluar, controlar y monitorear el rendimiento, impulsando mejoras de acuerdo a técnicas y normas vigentes definidas por los organismos de estandarización.	No aporta
CE.4.1. Certificar el funcionamiento, condición de uso o estado de sistemas de información, sistemas de comunicación de datos, software, seguridad informática y calidad de software para asegurar la generación de los resultados deseados en función de restricciones de tiempo y recursos establecidos.	No aporta
CE.5.1. Dirigir y controlar la implementación, operación y mantenimiento de sistemas de información, sistemas de comunicación de datos, software, seguridad informática y calidad de software, a los fines de alcanzar los objetivos fijados por la organización.	No aporta
CE.6.1. Asesorar y capacitar a organizaciones, empresas, organismos públicos o privados en la adquisición, instalación y uso, en lo que respecta a sistemas de información, sistemas de comunicación de datos, software, seguridad informática y calidad de software, a los fines de un uso correcto de los sistemas intervinientes.	No aporta
CE.7.1. Realizar pericias, tasaciones y arbitrajes relacionados con su actividad profesional, respetando marcos normativos y jurídicos con el objeto de asesorar a las partes o a los tribunales de Justicia.	No aporta

#### 4. Contenidos Mínimos

- [1]. Programación Imperativa y Concepto de Algoritmo.
- [2]. Concepto de Dato.
- [3]. Tipos de Datos Simples.
- [4]. Tipos Abstractos de Datos.
- [5]. Estructuras de Control Básicas.
- [6]. Estrategias de Resolución de Problemas.
- [7]. Estructuras de Datos.
- [8]. Abstracciones con Procedimientos y Funciones.
- [9]. Estructuras de Datos Lineales y No Lineales.
- [10]. Algoritmos de Búsqueda, Recorrido y Ordenamiento.
- [11]. Archivos de Acceso Secuencial y Aleatorio.
- [12]. Recursividad.

## 5. Objetivos establecidos en el DC

- Identificar problemas algorítmicos.
- Comprender el proceso de desarrollo de software.
- Resolver problemas aplicando soluciones algorítmicas y estructuras de datos.

## 6. Resultados de aprendizaje

Los siguientes resultados de aprendizaje se promueven en el desarrollo de la asignatura

Identificador de RA	Redacción
RA1	Reconocer los fundamentos de programación para desarrollar un algoritmo secuencial simple, identificando datos, procesos y resultados en la resolución de problemas simples.
RA2	Seleccionar las estructuras de control necesarias para resolver problemas de índole condicional o repetitiva, considerando las características de cada problema.
RA3	Especificar algoritmos modulares, para poder obtener soluciones más pequeñas y reutilizables, aplicando procedimientos y funciones.
RA4	Seleccionar soluciones ya implementadas (algoritmos), para poder usarlas en diferentes problemas, aplicando estrategias diversas.
RA5	Reconocer una estructura de datos para usarla en problemas propuestos, aplicando diferentes técnicas algorítmicas.
RA6	Programar algoritmos con estructuras de datos en memoria temporal y permanente que puedan resolver diferentes problemas, utilizando arreglos y archivos respectivamente.
RA7	Seleccionar algoritmos con diferentes niveles de eficiencia para manejar grandes volúmenes de datos, utilizando algoritmos de búsquedas y ordenamiento.
RA8	Definir tipos de datos abstractos para representar objetos o entidades involucradas en un problema determinado, utilizando encapsulamiento de variables y funciones.
RA9	Programar algoritmos con estructuras de datos no lineales en memoria temporal que puedan resolver diferentes problemas, utilizando árboles y grafos.
RA10	Distinguir las estrategias de resolución de problemas y técnicas básicas de análisis de algoritmos, para resolver en forma eficiente problemas diversos, considerando en forma amplia todos los recursos incorporados en la asignatura.

## 7. Relación de los RA y las competencias

En la tabla siguiente se indica con X la tributación de cada Resultado de Aprendizaje con las competencias de egreso: específicas, genéricas tecnológicas, sociales, políticas y actitudinales de la carrera.

RA	CE1.1	CE1.2	CE1.3	CE2.1	CE3.1	CE4.1	CE5.1	CE6.1	CE7.1	CG1	CG2	CG3	CG4	CG5	CG6	CG7	CG8	CG9	CG10	CG11
RA1		-	X	-	-	-	-	-	-	X	-	-	X	X	X	X	X	-	X	-
RA2		-	X	-	-	-	-	-	-	X	-	-	X	X	X	X	X	-	X	-
RA3		-	X	-	-	-	-	-	-	X	-	-	X	X	X	X	X	-	X	-
RA4		-	X	-	-	-	-	-	-	X	-	-	X	X	X	X	X	-	X	-
RA5		-	X	-	-	-	-	-	-	X	-	-	X	X	X	X	X	-	X	-
RA6		-	X	-	-	-	-	-	-	X	-	-	X	X	X	X	X	-	X	-
RA7		-	X	-	-	-	-	-	-	X	-	-	X	X	X	X	X	-	X	-
RA8		-	X	-	-	-	-	-	-	X	-	-	X	X	X	X	X	-	X	-
RA9		-	X	-	-	-	-	-	-	X	-	-	X	X	X	X	X	-	X	-
RA10		-	X	-	-	-	-	-	-	X	-	-	X	X	X	X	X	-	X	-

## 8. Asignaturas correlativas previas

Para cursar y rendir debe tener cursadas:

- Asignatura/s:  
Ninguna (salvo curso de ingreso)

Para cursar y rendir debe tener aprobada:

- Asignatura/s:  
Ninguna (salvo curso de ingreso)

## 9. Asignaturas correlativas posteriores

Indicar las asignaturas correlativas posteriores:

- Asignatura/s:  
Requieren AED CURSADA:  
Paradigmas de Programación  
Sintaxis y Semántica de los Lenguajes  
Análisis de Sistemas de Información  
Requieren AED APROBADA:  
Bases de Datos  
Desarrollo de Software  
Diseño de Sistemas de Información

## 10. Programa analítico

Este programa analítico contempla los contenidos mínimos, previstos en el DC vigente, y aquellos que se consideran necesarios para desarrollar los resultados de aprendizaje propuestos.

### **Unidad Nro. 1: Fundamentos de programación – Estructuras Secuenciales.**

#### **Contenidos.**

Los números que figuran entre corchetes hacen referencia a los contenidos mínimos previstos por la Ordenanza 1877/22, que están siendo efectivamente cubiertos por el ítem nombrado y resaltado en azul:

*Concepto de algoritmo* [1], computadora (o máquina algorítmica) y programa. Memoria, celdas de memoria y representación de información en base al sistema binario. Lenguajes de programación. *Concepto de dato* [2] y tipos de datos. *Tipos de datos simples* [3] en Python. Variables. Instrucciones básicas: asignación, carga por teclado y visualización por consola estándar. Operadores aritméticos. Expresiones aritméticas (o numéricas). *Tipos estructurados* [7] elementales en Python: secuencias de datos y cadenas de caracteres. *Estructuras de control*

*de flujo básicas: la estructura secuencial de instrucciones* [5]. Concepto de script en Python. Estructura de un script, edición, depuración y ejecución a través de la consola de comandos de Python y a través de un IDE. Reglas y convenciones de escritura de código fuente del Lenguaje Python. Uso de comentarios en un script Python. Pasos en la resolución de problemas y *estructura de un algoritmo* [1]. Técnicas de representación de algoritmos: pseudocódigo y diagrama de flujo. Definición de problema simple. Planteo y resolución de problemas simples en base a *secuencias de instrucciones simples* [5].

**Carga horaria:** 15 horas cátedra (3 semanas).

## **Unidad Nro. 2: Estructuras Condicionales - Subproblemas.**

### **Contenidos.**

Los números que figuran entre corchetes hacen referencia a los contenidos mínimos previstos por la Ordenanza 1877/22, que están siendo efectivamente cubiertos por el ítem nombrado y resaltado en azul:

*Estructuras de control de flujo básicas: la instrucción condicional* [5]. *Instrucción condicional simple, doble y múltiple en Python* [5]. Operadores relacionales y lógicos. Expresiones y proposiciones lógicas (o booleanas). Tablas de verdad. Condiciones anidadas. Planteo de problemas simples en base a *instrucciones condicionales* [5]. Concepto de Subproblema. Problemas compuestos (divisibles en subproblemas). Descomposición de un problema compuesto en subproblemas simples. Planteo y resolución de problemas compuestos en base a *instrucciones condicionales* [5].

**Carga horaria:** 10 horas cátedra (2 semanas).

## **Unidad Nro. 3: Estructuras Repetitivas.**

### **Contenidos.**

Los números que figuran entre corchetes hacen referencia a los contenidos mínimos previstos por la Ordenanza 1877/22, que están siendo efectivamente cubiertos por el ítem nombrado y resaltado en azul:

Concepto de proceso repetitivo. *Estructuras de control de flujo básicas: la estructura cíclica o repetitiva* [5]. Tipos generales de ciclos: el ciclo 0-N y el ciclo 1-N. Los ciclos while y for en Python. Ciclos infinitos. Interrupción de un ciclo. Ciclos anidados. Variables de conteo y de acumulación. Variables centinela (o banderas). Planteo y resolución de problemas de naturaleza repetitiva básica usando *instrucciones cíclicas* [5]. Carga y procesamiento de sucesiones de datos desde la consola estándar, conociendo la cantidad de datos o mediante un proceso de carga por doble lectura. Determinación de promedios y porcentajes. Determinación del mayor o menor de una sucesión de valores. Generación de números aleatorios y su aplicación en juegos y simulaciones simples. Procesamiento básico de caracteres y cadenas de caracteres.

**Carga horaria:** 20 horas (4 semanas).

## **Unidad Nro. 4: Subrutinas y Funciones.**

### **Contenidos.**

Los números que figuran entre corchetes hacen referencia a los contenidos mínimos previstos por la Ordenanza 1877/22, que están siendo efectivamente cubiertos por el ítem nombrado y resaltado en azul:

*Concepto elemental de subrutina para plantear un subproblema: introducción al uso de funciones [8]. Funciones en Python: cabecera y estructura general. Funciones con retorno de valor y funciones sin retorno de valor. Pasaje de parámetros a una función [6]. Parámetros actuales y parámetros formales. Formas especiales de parametrización en Python. Concepto general/teórico de parametrización por valor y por referencia y el modelo en Python: parametrización por valor. Ámbito de una variable: variables locales y variables globales a un bloque de acciones. Noción de programación imperativa (o por procedimientos) [1] y programación estructurada [1]: planteo de un programa en forma modular usando funciones. Generalización de código fuente: planteo de funciones parametrizadas de uso general [8]. Reutilización de código. Módulos y paquetes de módulos en Python. Módulos de la librería estándar más comunes en Python. Importación de módulos. Creación de módulos y paquetes del programador. Recursividad [12]. Funciones recursivas bien planteadas [12]. Determinación de la condición de corte de la recursividad. Seguimiento de la recursividad. Análisis básico de la eficiencia y la aplicabilidad de la recursividad en situaciones prácticas comunes.*

**Carga horaria:** 25 horas cátedra (5 semanas).

### **Unidad Nro. 5: Arreglos.**

#### **Contenidos.**

Los números que figuran entre corchetes hacen referencia a los contenidos mínimos previstos por la Ordenanza 1877/22, que están siendo efectivamente cubiertos por el ítem nombrado y resaltado en azul:

Concepto de tipo compuesto o estructurado (estructuras de datos vs. variables de tipo simple) *Estructuras de datos [7]:* definición general y necesidades de uso. *Arreglos [7]* (variables subindicadas): concepto de acceso directo y acceso secuencial. Dimensión de un arreglo. Índices. Implementación del concepto de arreglo de tamaño dinámico mediante listas nativas en Python. Arreglos unidimensionales (o *vectores*): creación y asignación de valores. *Recorrido secuencial de un vector [10]*. Carga y visualización. Vectores de conteo y vectores de acumulación. Vectores paralelos. Arreglos bidimensionales (o *matrices*) y multidimensionales: conceptos y aplicaciones. Creación de matrices y asignación de valores. *Recorrido de matrices en orden de filas y en orden de columnas [10]*. Sumarización por filas y columnas.

**Carga horaria:** 25 horas cátedra (5 semanas).

### **Unidad Nro. 6: Ordenamiento y Búsqueda.**

#### **Contenidos.**

Los números que figuran entre corchetes hacen referencia a los contenidos mínimos previstos por la Ordenanza 1877/22, que están siendo efectivamente cubiertos por el ítem nombrado y resaltado en azul:

*Ordenamiento: clasificación tradicional en métodos directos (o simples) y compuestos (o mejorados). Algoritmos simples o directos (Bubble sort, Selection sort e Insertion sort) [10]. Algoritmos mejorados o compuestos (Shellsort, Heapsort, Quicksort) [10]. Algoritmos de Búsqueda Secuencial y Búsqueda Binaria [10]. Fusión (o mezcla) de vectores ordenados. Fundamentos de Análisis de Algoritmos. Factores básicos: tiempo de ejecución y espacio de memoria empleado. Casos de análisis: peor caso, mejor caso, caso promedio. Concepto de comportamiento asintótico. Concepto de Orden de Complejidad. Notación *O mayúscula* (o *Big O*). Otras notaciones (Omega, Theta, Little o). Análisis de eficiencia de los distintos algoritmos de ordenamiento y búsqueda presentados en la unidad.*

**Carga horaria:** 15 horas cátedra (3 semanas).

## **Unidad Nro. 7: Registros. Tipos de Datos Abstractos.**

### **Contenidos.**

Los números que figuran entre corchetes hacen referencia a los contenidos mínimos previstos por la Ordenanza 1877/22, que están siendo efectivamente cubiertos por el ítem nombrado y resaltado en azul:

Concepto clásico de registro. Creación básica de registros en Python. Campos y acceso a campos. Noción elemental de Programación Orientada a Objetos (POO): definición y conceptos básicos. Encapsulamiento. Clases e instancias/objetos. Atributos y métodos. Creación de clases en Python. Función constructora. Conversión a cadena. Acceso a atributos y métodos. *Concepto de abstracción de datos y abstracción funcional: empleo de clases para modelarlos [4] [8]*. Tipos nativos de datos y *tipos abstractos de datos [4]*. Uso combinado de arreglos y objetos: vectores y matrices de objetos. *Implementación de tipos abstractos [8]* que representen números complejos, puntos en un plano y entidades del dominio para sistemas de gestión administrativa (como clientes, cuentas, artículos, libros y otras).

**Carga horaria:** 10 horas cátedra (2 semanas).

## **Unidad Nro. 8: Estructuras de Datos Lineales y No Lineales.**

### **Contenidos.**

Los números que figuran entre corchetes hacen referencia a los contenidos mínimos previstos por la Ordenanza 1877/22, que están siendo efectivamente cubiertos por el ítem nombrado y resaltado en azul:

Estructuras de datos nativas y *estructuras de datos abstractas [4] [7]*. *Conceptos de estructuras de datos lineales y no lineales [7]*. *Estructuras lineales básicas: Estructuras tipo LIFO (pilas) y tipo FIFO (colas) [7]*. *Listas [7]*. Conceptos asociados. Diseño de clases para la implementación de pilas y colas usando arreglos adaptativos como soporte. *Estructuras no lineales básicas: árboles binarios y grafos [7]*. Árboles binarios de búsqueda: conceptos asociados y diseño de clases para su implementación elemental (sin control de equilibrio). Grafos: conceptos asociados y diseño de clases para su implementación elemental en forma matricial.

**Carga horaria:** 10 horas cátedra (2 semanas).

## **Unidad Nro. 9: Archivos.**

### **Contenidos.**

Los números que figuran entre corchetes hacen referencia a los contenidos mínimos previstos por la Ordenanza 1877/22, que están siendo efectivamente cubiertos por el ítem nombrado y resaltado en azul:

El problema de la volatilidad de los datos en memoria principal. Concepto y necesidad de medios de almacenamiento externo. *Archivos [11]*. Archivos de texto y binarios. Apertura. Modos de apertura. Lectura. Escritura. Final de archivo. Reposicionamiento. *Formas de organización y accesos: archivos de acceso secuencial y archivos de acceso aleatorio [11]*. Elementos de procesamiento básico de archivos de registros/objetos (y archivos binarios en general): altas, bajas (lógicas y físicas) y modificaciones (procesamiento *ABM*), listados de contenido (completos o filtrados), búsqueda secuencial, consultas. Archivos de texto como soporte de lotes de datos secuenciales para su uso en problemas de procesamiento masivo de datos. Creación, carga,

recorrido, conversión de contenidos (de tipo texto a tipo numérico) y eventual creación de arreglos a partir de esos contenidos.

**Carga horaria:** 20 horas cátedra (4 semanas).

### **Unidad Nro. 10: Estrategias de Resolución de Problemas.**

#### **Contenidos.**

Los números que figuran entre corchetes hacen referencia a los contenidos mínimos previstos por la Ordenanza 1877/22, que están siendo efectivamente cubiertos por el ítem nombrado y resaltado en azul:

Motivación y antecedentes para el estudio de *estrategias de resolución de problemas (o de planteo de algoritmos)* [6]. Clasificación y concepto básico de las estrategias existentes más comunes: fuerza bruta, recursividad, backtraking, divide y vencerás, randomización, algoritmos ávidos y programación dinámica. Revisión de planteos algorítmicos conocidos basados en Fuerza Bruta. *Recursividad* [12] y *Backtraking*: planteo de algoritmos basados en aprovechar el proceso de vuelta atrás de la recursión para analizar soluciones parciales a un problema. Divide y Vencerás: estructura general de la estrategia.

**Carga horaria:** 10 horas cátedra (2 semanas).

### **Carga horaria por tipo de formación práctica de toda la asignatura**

<b>Tipo de formación práctica</b>	<b>Horas reloj</b>
Formación experimental	0
Análisis y resolución de problemas de ingeniería y estudios de casos	80
Formulación, análisis y desarrollo de proyectos.	0

### **Bibliografía Obligatoria:**

- Frittelli, V., et al. (2022). Publicaciones Cátedra de AED. *Fichas de Estudio para AED (disponible en Aula Virtual AED)*. Córdoba: UTN – Facultad Regional Córdoba.
- Python Software Foundation. (2022). *Python Documentation*. Retrieved August 30, 2022, from <https://docs.python.org/3/>.

### **Bibliografía optativa y otros materiales a utilizar en la asignatura:**

- Pilgrim, M. (2004). *Dive Into Python - Python from novice to pro*. New York: Apress.
- Frittelli, V., et al (2020-2022). Videos específicos para la Cátedra de AED (disponibles en Aula Virtual AED). Córdoba: UTN – Facultad Regional Córdoba.

## **11. Metodología de enseñanza**

- ✓ En cada clase se desarrolla un tema central, y el mismo se ejemplifica y analiza con modelos de programas presentados por los profesores.

- ✓ Los alumnos realizan modificaciones y variantes sobre los modelos, y realizan ejercicios nuevos en base a los temas tratados.
- ✓ Se utilizan de manera regular y continua diversos foros de consulta, en los cuales los estudiantes plantean dudas o requerimientos diversos, que son respondidos por los docentes de la cátedra o incluso por otros estudiantes.
- ✓ Los trabajos prácticos y las tareas semanales (cuestionarios y desafíos) integran los conocimientos de las distintas unidades, y parte del tiempo de clase se usa para analizar dudas y elementos relevantes referidos a esos trabajos.
- ✓ Todos los materiales del curso (materiales del estudio y evaluaciones de tipos diversos) están siempre disponibles en el aula virtual de la asignatura.
- ✓ Se dispone también de guías de problemas, que son enunciados específicos referidos temas de cada semana o tema tratado, para los que se anexan las soluciones sugeridas por la Cátedra. Las soluciones se hacen visibles a los estudiantes al finalizar cada semana de cursado, para favorecer que intenten encontrar las soluciones sin tener a la vista las sugerencias de sus docentes.

## 12. Recomendaciones para el estudio

- Cada estudiante debe instalar en su computadora personal las herramientas de software requeridas por la materia, y realizar en forma personal prácticas y pruebas sobre ellas para lograr dominio pleno de su uso.
- Dedicar tiempo extra (en sus casas y por fuera de las horas regulares de clase) para estudio y aplicación práctica, equivalente al menos a la misma cantidad de horas cátedra que tiene la asignatura. Al menos 5 horas cátedra extra por semana, dedicadas a estudio y aplicación de contenidos.
- Dentro de ese tiempo extra, contemplar la lectura de los materiales disponibles cada semana, y la realización del cuestionario teórico asociado a los materiales de cada semana.
- Cada estudiante debe realizar en forma personal prácticas y pruebas sobre los distintos temas de las unidades del programa de la asignatura, para lograr dominio pleno de su uso.
- Si en la semana aparece alguna actividad adicional (como un desafío de programación) contemplar en ese mismo tiempo extra el diseño de soluciones para el mismo, programar y probar esas soluciones.
- Publicar en los foros habilitados por la cátedra las dudas que vayan surgiendo, y colaborar con las respuestas a las dudas de otros compañeros.
- Integrar un grupo de estudios regular durante el cursado, con estudiantes que se perciban en un nivel de conocimientos similar, y reunir ese grupo para el desarrollo de las actividades que surjan.
- No quedarse con dudas en el transcurso de las clases: preguntar y consultar a sus profesores en el momento en que la duda surja, o al final de la clase.
- Estudiar las soluciones que se propongan desde la cátedra a los diversos ejercicios y problemas que se propongan, y probar a modificar esos programas para producir otros resultados o variantes de alguno ya obtenido.

- No quedarse atrás en el estudio de los elementos conceptuales que van surgiendo semana tras semana en el cursado.
- Por cualquier inconveniente o duda que no pueda resolverse mediante las vías normales ya descriptas, siempre está disponible la alternativa de contactar en forma directa a los profesores, mediante canales de comunicación disponibles en el aula virtual, o en forma presencial en la Facultad.

### 13. Metodología de evaluación

El modelo de enseñanza basado en competencias implica la aplicación de metodologías e instrumentos de evaluación que permiten conocer, a docentes y estudiantes, el nivel de desarrollo de las competencias que aborda la asignatura.

En todos los cursos de la cátedra se aplica un criterio único, igual para todos los estudiantes, basado en evaluación continua y progresiva.

Cada semana se habilitan Cuestionarios Teóricos cuyo objetivo es favorecer que los estudiantes revisen y afiancen los contenidos que se desarrollaron en cada semana y avancen en forma regular y sostenida en sus estudios.

Se dispone también de actividades de programación designadas como Desafíos de Programación, que implican el desarrollo de soluciones para situaciones problemáticas exigentes, debiendo informar los resultados obtenidos (sin necesidad de subir los programas).

También se solicitan Trabajos Prácticos a desarrollar en forma individual o en forma grupal (según se indique en las consignas) para diseñar soluciones a problemas situacionales exigentes, debiendo entregar los programas para su evaluación. Está disponible una instancia de Trabajo Práctico Recuperatorio para uno de estos Trabajos Prácticos.

Están previstos cuatro Parciales en momentos clave del año para evaluar en forma individual el nivel de avance de cada estudiante frente a problemas que debe desarrollar en un ambiente controlado y en tiempo acotado. Si es necesario, el estudiante puede hacer un Parcial Recuperatorio para uno de estos cuatro parciales.

Finalmente, se prevé un quinto parcial (designado como Parcial 5) solo para evaluar en forma general los contenidos de las últimas unidades de la materia, que no se evalúan en parciales ni en prácticos. Todos los estudiantes deben hacer este quinto parcial, que prevé la posibilidad de dos intentos (contando el segundo como recuperatorio en caso de ser necesario).

Todas estas evaluaciones apuntan a la formación en cada uno de los resultados de aprendizaje definidos, que a su vez aportan a las competencias específicas y generales indicadas para la asignatura.

A continuación, se detallan todos los Resultados de Aprendizajes con sus contenidos a desarrollar para alcanzarlos, la mediación pedagógica, metodologías y estrategias de evaluación, tiempo en horas reloj.

Resultados de Aprendizaje	Contenidos según programa	Mediación Pedagógica	Metodología y Estrategias de Evaluación	Tiempos en hora reloj
<p>RA 1: Reconocer los fundamentos de programación para desarrollar un algoritmo secuencial simple, identificando datos, procesos y resultados en la resolución de problemas simples.</p>	<p><b>Fundamentos de programación – Estructuras secuenciales.</b> Concepto de algoritmo, computadora y programa. Memoria. Lenguajes de programación. Concepto de dato y tipos de datos. Variables. Instrucciones básicas. La estructura secuencial de instrucciones. Técnicas de representación de algoritmos. Definición de problema simple.  (Ver desglose y detalle de temas en ítem 10 – Programa Analítico [Unidad 1]).</p>	<p><b>Estrategias:</b> Desarrollo de temas en forma de lección magistral participativa. Planteo de preguntas guía. Planteo de ejercicios y problemas. Fomentar el aprendizaje colaborativo mediante el trabajo en grupos. Propuesta de estudio de casos amplios.  <b>Actividades:</b> Proponer respuestas a temas y preguntas guía que se planteen en el desarrollo de cada clase, referidas a planteo de algoritmos basados en estructuras secuenciales. Resolver los ejercicios y problemas, en forma individual o en forma grupal (según sea requerido). Plantear soluciones genéricas mediante esquemas o respuestas planteadas en papel o en documentos digitales, tales que requieran especialmente identificar</p>	<p><b>Instrumentos:</b> Cuestionarios planteados sobre los contenidos presentados. Problema de programación en modalidad de desafío: procesar un lote de datos concreto e informar los resultados. Problemas de complejidad incremental sugeridos en las guías de problemas propuestas en el curso. Trabajo práctico grupal integrador (resolución por etapas a lo largo del cursado). Examen Parcial a desarrollar en el laboratorio y en forma individual.  <b>Criterios Generales:</b> Participa en forma activa en el desarrollo de cada clase, aportando ideas claras y soluciones bien expresadas,</p>	<p>Horas presenciales áulicas: 10 Horas teóricas: 5 Horas prácticas: 5 (Horas extra áulicas: 5)</p>

		<p>datos, procesos básicos y resultados en cada problema planteado.          Plantear soluciones específicas, con estructura de algoritmo secuencial simple, en computadora mediante el lenguaje de programación propuesto.          Sugerir el planteo de consultas y dudas en los foros del curso, y la participación en la respuesta a esas consultas.          Analizar variantes de solución para los casos amplios propuestos.</p>	<p>pertinentes y entendibles tanto en expresiones coloquiales como en lenguaje y comunicación coloquial.          Propone alternativas de solución para desafíos y problemas, mediante explicaciones adecuadas con justificación lógica y correctamente expresadas.          Responde los cuestionarios en tiempo y forma.</p> <p><b>Criterios Específicos:</b>          Participa en forma activa en el planteo de soluciones con base en estructuras secuencias para los ejercicios sugeridos en las guías y en el trabajo práctico integrador.          Resuelve correctamente cada aspecto abordado en en cada problema o evaluado en el parcial, distinguiendo datos, procesos y resultados y planteando un algoritmo</p>	
--	--	--	--	--

			simple acorde al nivel requerido.	
<p>RA 2</p> <p>Seleccionar las estructuras de control necesarias para resolver problemas de índole condicional o repetitiva, considerando las características de cada problema.</p>	<p><b>Estructuras Condicionales - Subproblemas.</b> La instrucción condicional. Operadores relacionales y lógicos. Condiciones anidadas. Planteo de problemas simples en base a instrucciones condicionales. Concepto de Subproblema. Problemas compuestos. Planteo y resolución de problemas compuestos en base a instrucciones condicionales.</p> <p><b>Estructuras Repetitivas.</b> La estructura repetitiva. Tipos generales de ciclos. Ciclos infinitos. Planteo y resolución de problemas de naturaleza repetitiva básica usando instrucciones cíclicas.</p> <p>(Ver desglose y detalle de temas en el Ítem 10 – Programa Analítico [Unidades 2 y 3]).</p>	<p><b>Estrategias:</b> Desarrollo de temas en forma de lección magistral participativa. Planteo de preguntas guía. Planteo de ejercicios y problemas que incorporen naturaleza condicional o naturaleza repetitiva. Fomentar el aprendizaje colaborativo mediante el trabajo en grupos. Propuesta de estudio de casos amplios.</p> <p><b>Actividades:</b> Proponer respuestas a temas y preguntas guía que se planteen en el desarrollo de cada clase. Resolver los ejercicios y problemas, en forma individual o en forma grupal (según sea requerido). Plantear soluciones genéricas mediante pseudocódigo o diagramas de flujo, que impliquen uso de estructuras condicionales y repetitivas. Plantear soluciones específicas en computadora mediante el lenguaje de programación propuesto.</p>	<p><b>Instrumentos:</b> Cuestionarios planteados sobre los contenidos presentados. Problema de programación en modalidad de desafío: procesar un lote de datos concreto e informar los resultados. Problemas de complejidad incremental sugeridos en las guías de problemas propuestas en el curso. Trabajo práctico grupal integrador (resolución por etapas a lo largo del cursado). Examen Parcial a desarrollar en el laboratorio y en forma individual.</p> <p><b>Criterios Generales:</b> Participa en forma activa en el desarrollo de cada clase, aportando ideas claras y soluciones bien expresadas, pertinentes y entendibles tanto en expresiones coloquiales como en</p>	<p>Horas presenciales áulicas: 20</p> <p>Horas teóricas: 10</p> <p>Horas prácticas: 10</p> <p>(Horas extra áulicas: 10)</p>

		<p>Adequar soluciones basadas en estructuras secuenciales, para transformarlas en soluciones basadas en estructuras repetitivas. Sugerir el planteo de consultas y dudas en los foros del curso, y la participación en la respuesta en esos mismos foros a esas consultas. Analizar variantes de solución para los casos amplios propuestos.</p>	<p>lenguaje y comunicación coloquial. Propone alternativas de solución para desafíos y problemas, mediante explicaciones adecuadas con justificación lógica y correctamente expresadas. Responde los cuestionarios en tiempo y forma.</p> <p><b>Criterios Específicos:</b>          Propone alternativas de solución para desafíos y problemas, basadas en diversas forma de uso de condiciones y ciclos. Muestra dominio en el planteo de soluciones de naturaleza repetitiva. Participa en forma activa en el planteo de soluciones condicionales y cíclicas para cada problema sugerido en guás y para el trabajo práctico integrador. Resuelve correctamente mediante estructuras condicionales o cíclicas cada aspecto evaluado en los problemas sugeridos y el parcial.</p>	
--	--	--	---	--

<p>RA 3</p> <p>Especificar algoritmos modulares, para poder obtener soluciones más pequeñas y reutilizables, aplicando procedimientos y funciones.</p>	<p><b>Subrutinas y Funciones.</b></p> <p>Concepto elemental de subrutina. Funciones. Retorno de valor. Pasaje de parámetros. Ámbito de una variable: variables locales y globales. Programación imperativa y programación estructurada.</p> <p>(Ver desglose y detalle de temas en el Ítem 10 – Programa Analítico [Unidad 4]).</p>	<p><b>Estrategias:</b></p> <p>Desarrollo de temas en forma de lección magistral participativa. Planteo de preguntas guía. Planteo de ejercicios y problemas que incorporen el desarrollo de funciones separadas. Fomentar el aprendizaje colaborativo mediante el trabajo en grupos. Propuesta de estudio de casos amplios.</p> <p><b>Actividades:</b></p> <p>Proponer respuestas a temas y preguntas guía que se planteen en el desarrollo de cada clase. Resolver los ejercicios y problemas, en forma individual o en forma grupal (según sea requerido). Plantear soluciones genéricas mediante esquemas o respuestas planteadas en papel o en documentos digitales. Plantear soluciones específicas en computadora mediante el lenguaje de programación propuesto. Proponer soluciones basadas en el planteo de funciones separadas. Sugerir el planteo de consultas y dudas en los foros del curso, y la</p>	<p><b>Instrumentos:</b></p> <p>Cuestionarios planteados sobre los contenidos presentados. Problema de programación en modalidad de desafío: procesar un lote de datos concreto e informar los resultados. Problemas de complejidad incremental sugeridos en las guías de problemas propuestas en el curso. Trabajo práctico grupal integrador (resolución por etapas a lo largo del cursado). Examen Parcial a desarrollar en el laboratorio y en forma individual.</p> <p><b>Criterios Generales:</b></p> <p>Participa en forma activa en el desarrollo de cada clase, aportando ideas claras y soluciones bien expresadas, pertinentes y entendibles tanto en expresiones coloquiales como en lenguaje y comunicación coloquial.</p>	<p>Horas presenciales áulicas: 10</p> <p>Horas teóricas: 6</p> <p>Horas prácticas: 4</p> <p>(Horas extra áulicas: 5)</p>
--	---	--	--	--

		participación en la respuesta en esos mismos foros a esas consultas. Analizar variantes de solución (basadas en el uso de funciones) para los casos amplios propuestos.	<p>Propone alternativas de solución para desafíos y problemas, mediante explicaciones adecuadas con justificación lógica y correctamente expresadas. Responde los cuestionarios en tiempo y forma.</p> <p><b>Criterios Específicos:</b> Muestra dominio en el planteo de funciones para el desarrollo de programas. Participa en forma activa en el planteo de soluciones basadas en uso de funciones separadas (reutilizables o no) para resolver cada problema sugerido y el trabajo práctico integrador. Resuelve correctamente cada aspecto evaluado en el parcial, incluyendo uso de funciones separadas para modularizar esa solución.</p>	
RA 4 Seleccionar soluciones ya implementadas	<b>Subrutinas y Funciones.</b> Planteo de programas en forma modular usando funciones. Generalización de código fuente. Reutilización de código. Módulos y paquetes.	<b>Estrategias:</b> Desarrollo de temas en forma de lección magistral participativa. Planteo de preguntas guía. Planteo de ejercicios y problemas que incorporen el reuso de funciones separadas.	<b>Instrumentos:</b> Cuestionarios planteados sobre los contenidos presentados. Problema de programación en modalidad de desafío: procesar un lote de datos	Horas presenciales áulicas: 10 Horas teóricas: 6 Horas prácticas: 4 (Horas extra áulicas: 5)

<p>(algoritmos), para poder usarlas en diferentes problemas, aplicando estrategias diversas.</p>	<p>(Ver desglose y detalle de temas en el Ítem 10 – Programa Analítico [Unidad 4]).</p>	<p>Planteo de ejercicios y problemas que incorporen el diseño de módulos conteniendo funciones reutilizables. Fomentar el aprendizaje colaborativo mediante el trabajo en grupos. Propuesta de estudio de casos amplios.</p> <p><b>Actividades:</b> Proponer respuestas a temas y preguntas guía que se planteen en el desarrollo de cada clase. Resolver los ejercicios y problemas, en forma individual o en forma grupal (según sea requerido). Plantear soluciones genéricas mediante esquemas lógicos que impliquen el uso de subrutinas separadas. Plantear soluciones específicas en computadora mediante el lenguaje de programación propuesto. Proponer soluciones basadas en el reuso de funciones y el diseño de módulos separados. Sugerir el planteo de consultas y dudas en los foros del curso, y la participación en la respuesta en esos mismos foros a esas consultas.</p>	<p>concreto e informar los resultados. Problemas de complejidad incremental sugeridos en las guías de problemas propuestas en el curso. Trabajo práctico grupal integrador (resolución por etapas a lo largo del cursado). Examen Parcial a desarrollar en el laboratorio y en forma individual.</p> <p><b>Criterios Generales:</b> Participa en forma activa en el desarrollo de cada clase, aportando ideas claras y soluciones bien expresadas, pertinentes y entendibles tanto en expresiones coloquiales como en lenguaje y comunicación coloquial. Propone alternativas de solución para desafíos y problemas, mediante explicaciones adecuadas con justificación lógica y correctamente expresadas. Responde los cuestionarios en tiempo y forma.</p>	
--	---	--	--	--

		Analizar variantes de solución (basadas en el uso de funciones) para los casos amplios propuestos.	<p><b>Criterios Específicos:</b></p> <p>Muestra dominio en el planteo y uso de módulos separados y en el reuso de funciones para el desarrollo de programas.</p> <p>Participa en forma activa en el planteo de soluciones basadas en funciones y librerías de funciones para resolver problemas diversos y para el trabajo práctico integrador.</p> <p>Resuelve correctamente cada aspecto evaluado en el parcial, aplicando división en funciones correctamente parametrizadas y manejo correcto del retorno de resultados.</p> <p>Usa correctamente módulos de funciones agrupadas en el planteo de programas.</p>	
RA 5 Reconocer una estructura de datos para usarla en problemas	<b>Subrutinas y Funciones.</b> Recursividad. Funciones recursivas bien planteadas. Seguimiento de la recursividad. Análisis básico de la eficiencia y la aplicabilidad de la recursividad en situaciones prácticas comunes.	<b>Estrategias:</b> Desarrollo de temas en forma de lección magistral participativa. Planteo de preguntas guía. Planteo de ejercicios y problemas que requieran seleccionar y aplicar diversas estructuras de datos.	<b>Instrumentos:</b> Cuestionarios planteados sobre los contenidos presentados. Problema de programación en modalidad de desafío: procesar un lote de datos	Horas presenciales áulicas: 23 Horas teóricas: 13 Horas prácticas: 10 (Horas extra áulicas: 12)

<p>propuestos, aplicando diferentes técnicas algorítmicas.</p>	<p><b>Arreglos.</b> Estructuras de datos. Arreglos. Concepto de acceso directo y acceso secuencial. Índices. Arreglos unidimensionales. Arreglos bidimensionales.</p> <p><b>Registros.</b> Concepto clásico de registro. Campos y acceso a campos. Noción elemental de Programación Orientada a Objetos (POO). Encapsulamiento. Clases e instancias/objetos. Atributos y métodos. Función constructora. Uso combinado de arreglos y objetos.</p> <p><b>Estructuras de Datos Lineales y No Lineales.</b> Estructuras lineales básicas: Tipo LIFO (pilas) y tipo FIFO (colas). Listas. Estructuras no lineales básicas: Árboles binarios de búsqueda (sin control de equilibrio) y Grafos (implementación elemental en forma matricial).</p> <p><b>Archivos.</b> Archivos de texto y binarios. Apertura. Lectura. Escritura. Final de archivo. Reposicionamiento.</p>	<p>Planteo de ejercicios y problemas que requieran seleccionar y aplicar diversas estrategias de planteo de algoritmos. Fomentar el aprendizaje colaborativo mediante el trabajo en grupos. Propuesta de estudio de casos amplios.</p> <p><b>Actividades:</b> Proponer respuestas a temas y preguntas guía que se planteen en el desarrollo de cada clase. Resolver los ejercicios y problemas, en forma individual o en forma grupal (según sea requerido). Plantear soluciones genéricas mediante esquemas o respuestas planteadas en papel o en documentos digitales. Plantear soluciones mediante el lenguaje de programación propuesto, basadas en seleccionar y usar en forma correcta las estructuras de datos que correspondan a cada caso. Proponer soluciones en la aplicación de diferentes estrategias (recursivas o no recursivas) de planteo de algoritmos.</p>	<p>concreto e informar los resultados. Problemas de complejidad incremental sugeridos en las guías de problemas propuestas en el curso. Trabajo práctico grupal integrador (resolución por etapas a lo largo del cursado). Examen Parcial a desarrollar en el laboratorio y en forma individual.</p> <p><b>Criterios Generales:</b> Participa en forma activa en el desarrollo de cada clase, aportando ideas claras y soluciones bien expresadas, pertinentes y entendibles tanto en expresiones coloquiales como en lenguaje y comunicación coloquial. Propone alternativas de solución para desafíos y problemas, mediante explicaciones adecuadas con justificación lógica y correctamente expresadas. Responde los cuestionarios en tiempo y forma.</p>	
--	---	---	--	--

	<p>Archivos de acceso secuencial y archivos de acceso aleatorio.</p> <p><b>Estrategias de Resolución de Problemas.</b> Clasificación básica. Fuerza bruta, recursividad, backtraking, divide y vencerás, algoritmos ávidos y programación dinámica.</p> <p>(Ver desglose y detalle de temas en el Ítem 10 – Programa Analítico [Unidades 4, 5, 7, 8, 9 y 10]).</p>	<p>Sugerir el planteo de consultas y dudas en los foros del curso, y la participación en la respuesta en esos mismos foros a esas consultas. Analizar variantes de solución para los casos amplios propuestos.</p>	<p><b>Criterios Específicos:</b> Muestra dominio en el uso de estructuras de datos diversas para la solución de problemas. Selecciona en forma correcta las estrategias algorítmicas (con base recursiva o no recursiva) adecuadas para cada problema. Participa en forma activa en el planteo de soluciones basadas en diversas estructuras de datos para los problemas sugeridos en las guías o en el trabajo práctico integrador. Resuelve correctamente en forma iterativa o en forma recursiva cada problema propuesto en guías o evaluado en el parcial.</p>	
<p>RA 6</p> <p>Programar algoritmos con estructuras de datos en memoria</p>	<p><b>Arreglos.</b> Estructuras de datos. Arreglos. Concepto de acceso directo y acceso secuencial. Índices. Arreglos unidimensionales. Arreglos bidimensionales.</p> <p><b>Archivos.</b></p>	<p><b>Estrategias:</b> Desarrollo de temas en forma de lección magistral participativa. Planteo de preguntas guía. Planteo de ejercicios y problemas que requieran distinguir entre uso de memoria principal o uso de memoria externa.</p>	<p><b>Instrumentos:</b> Cuestionarios planteados sobre los contenidos presentados. Problema de programación en modalidad de desafío: procesar un lote de datos concreto e informar los resultados.</p>	<p>Horas presenciales áulicas: 16 Horas teóricas: 10 Horas prácticas: 6 (Horas extra áulicas: 6)</p>

<p>temporal y permanente que puedan resolver diferentes problemas, utilizando arreglos y archivos respectivamente.</p>	<p>Archivos de texto y binarios. Apertura. Lectura. Escritura. Final de archivo. Reposicionamiento. Archivos de acceso secuencial y archivos de acceso aleatorio.</p> <p>(Ver desglose y detalle de temas en el Ítem 10 – Programa Analítico [Unidades 5 y 10]).</p>	<p>Planteo de ejercicios y problemas que requieran usar correctamente estructuras de memoria principal y memoria externa.</p> <p>Fomentar el aprendizaje colaborativo mediante el trabajo en grupos.</p> <p>Propuesta de estudio de casos amplios.</p> <p><b>Actividades:</b></p> <p>Proponer respuestas a temas y preguntas guía que se planteen en el desarrollo de cada clase.</p> <p>Resolver los ejercicios y problemas, en forma individual o en forma grupal (según sea requerido).</p> <p>Proponer programas basados en la aplicación de técnicas de memoria principal, memoria externa, o combinaciones entre ambas.</p> <p>Sugerir el planteo de consultas y dudas en los foros del curso, y la participación en la respuesta en esos mismos foros a esas consultas.</p> <p>Analizar variantes de solución para los casos amplios propuestos.</p>	<p>Problemas de complejidad incremental sugeridos en las guías de problemas propuestas en el curso.</p> <p>Trabajo práctico grupal integrador (resolución por etapas a lo largo del cursado).</p> <p>Examen Parcial a desarrollar en el laboratorio y en forma individual.</p> <p><b>Criterios Generales:</b></p> <p>Participa en forma activa en el desarrollo de cada clase, aportando ideas claras y soluciones bien expresadas, pertinentes y entendibles tanto en expresiones coloquiales como en lenguaje y comunicación coloquial.</p> <p>Propone alternativas de solución para desafíos y problemas, mediante explicaciones adecuadas con justificación lógica y correctamente expresadas.</p> <p>Responde los cuestionarios en tiempo y forma.</p> <p><b>Criterios Específicos:</b></p>	
--	--	---	--	--

			<p>Aplica correctamente técnicas de manejo de datos en memoria principal y en memoria externa.</p> <p>Propone alternativas para el planteo de soluciones sobre memoria principal o sobre memoria secundaria para resolver los problemas propuestos y para el trabajo práctico integrador o para el parcial.</p>	
<p>RA 7</p> <p>Seleccionar algoritmos con diferentes niveles de eficiencia para manejar grandes volúmenes de datos, utilizando algoritmos de búsquedas y ordenamiento.</p>	<p><b>Ordenamiento y Búsqueda.</b></p> <p>Ordenamiento. Métodos directos: Bubble sort, Selection sort e Insertion sort. Métodos compuestos: Shellsort, Heapsort (noción), Quicksort. Búsqueda. Algoritmos de búsqueda secuencial y binaria.</p> <p>Fundamentos de Análisis de Algoritmos. Peor caso, mejor caso, caso promedio. Concepto de comportamiento asintótico. Concepto de Orden de Complejidad. Notación O mayúscula.</p> <p>(Ver desglose y detalle de temas en el ítem 10 – Programa Analítico [Unidad 6]).</p>	<p><b>Estrategias:</b></p> <p>Desarrollo de temas en forma de lección magistral participativa.</p> <p>Planteo de preguntas guía.</p> <p>Planteo de ejercicios y problemas que requieran aplicar técnicas de búsqueda y ordenamiento.</p> <p>Comparación básica entre los niveles de eficiencia de los distintos algoritmos disponibles.</p> <p>Fomentar el aprendizaje colaborativo mediante el trabajo en grupos.</p> <p>Propuesta de estudio de casos amplios.</p> <p><b>Actividades:</b></p> <p>Proponer respuestas a temas y preguntas guía que se planteen en el desarrollo de cada clase.</p>	<p><b>Instrumentos:</b></p> <p>Cuestionarios planteados sobre los contenidos presentados.</p> <p>Problema de programación en modalidad de desafío: procesar un lote de datos concreto e informar los resultados.</p> <p>Problemas de complejidad incremental sugeridos en las guías de problemas propuestas en el curso.</p> <p>Trabajo práctico grupal integrador (resolución por etapas a lo largo del cursado).</p> <p>Examen Parcial a desarrollar en el laboratorio y en forma individual.</p>	<p>Horas presenciales áulicas: 10</p> <p>Horas teóricas: 6</p> <p>Horas prácticas: 4</p> <p>(Horas extra áulicas: 3)</p>

		<p>Resolver los ejercicios y problemas, en forma individual o en forma grupal (según sea requerido).          Proponer soluciones basadas en la aplicación de distintas técnicas búsqueda y ordenamiento en contextos de volúmenes de datos crecientes.          Justificar las diferencias de rendimiento, base a técnicas elementales de análisis de algoritmos.          Sugerir el planteo de consultas y dudas en los foros del curso, y la participación en la respuesta en esos mismos foros a esas consultas.          Analizar variantes de solución para los casos amplios propuestos.</p>	<p><b>Criterios Generales:</b>          Participa en forma activa en el desarrollo de cada clase, aportando ideas claras y soluciones bien expresadas, pertinentes y entendibles tanto en expresiones coloquiales como en lenguaje y comunicación coloquial.          Propone alternativas de solución para desafíos y problemas, mediante explicaciones adecuadas con justificación lógica y correctamente expresadas.          Responde los cuestionarios en tiempo y forma.</p> <p><b>Criterios Específicos:</b>          Aplica correctamente diversas técnicas de búsqueda y ordenamiento, tanto en contextos controlados con volumen bajo de datos, como en situaciones realistas de volumen de datos masivo y creciente, tanto en ejercicios propuestos, como</p>	
--	--	--	--	--

			<p>el práctico integrador o en el parcial.</p> <p>Obtiene y justifica correctamente conclusiones sobre el rendimiento de los algoritmos usados.</p>	
<p>RA 8</p> <p>Definir tipos de datos abstractos para representar objetos o entidades involucradas en un problema determinado, utilizando encapsulamiento de variables y funciones.</p>	<p><b>Registros. Tipos de Datos Abstractos.</b></p> <p>Concepto clásico de registro. Campos y acceso a campos. Noción elemental de Programación Orientada a Objetos (POO). Encapsulamiento. Clases e instancias/objetos. Atributos y métodos. Función constructora. Uso combinado de arreglos y objetos.</p> <p>Abstracción de datos y abstracción funcional: empleo de clases para modelarlos. Tipos nativos de datos y tipos abstractos de datos.</p> <p>(Ver desglose y detalle de temas en el Ítem 10 – Programa Analítico [Unidad 7]).</p>	<p><b>Estrategias:</b></p> <p>Desarrollo de temas en forma de lección magistral participativa. Planteo de preguntas guía. Planteo de ejercicios y problemas que requieran definir e implementar nuevos tipos abstractos.</p> <p>Fomentar el aprendizaje colaborativo mediante el trabajo en grupos.</p> <p>Propuesta de estudio de casos amplios.</p> <p><b>Actividades:</b></p> <p>Proponer respuestas a temas y preguntas guía que se planteen en el desarrollo de cada clase.</p> <p>Resolver los ejercicios y problemas, en forma individual o en forma grupal (según sea requerido).</p> <p>Proponer soluciones basadas en la gestión de nuevos tipos, incorporando correctamente esos tipos en un programa modular.</p>	<p><b>Instrumentos:</b></p> <p>Cuestionarios planteados sobre los contenidos presentados.</p> <p>Problema de programación en modalidad de desafío: procesar un lote de datos concreto e informar los resultados.</p> <p>Problemas de complejidad incremental sugeridos en las guías de problemas propuestas en el curso.</p> <p>Trabajo práctico grupal integrador (resolución por etapas a lo largo del cursado).</p> <p>Examen Parcial a desarrollar en el laboratorio y en forma individual.</p> <p><b>Criterios Generales:</b></p> <p>Participa en forma activa en el desarrollo de cada clase, aportando ideas claras y</p>	<p>Horas presenciales áulicas: 7</p> <p>Horas teóricas: 4</p> <p>Horas prácticas: 3</p> <p>(Horas extra áulicas: 3)</p>

		<p>Sugerir el planteo de consultas y dudas en los foros del curso, y la participación en la respuesta en esos mismos foros a esas consultas. Analizar variantes de solución para los casos amplios propuestos.</p>	<p>soluciones bien expresadas, pertinentes y entendibles tanto en expresiones coloquiales como en lenguaje y comunicación coloquial.</p> <p>Propone alternativas de solución para desafíos y problemas, mediante explicaciones adecuadas con justificación lógica y correctamente expresadas. Responde los cuestionarios en tiempo y forma.</p> <p><b>Criterios Específicos:</b> Aplica correctamente técnicas básicas de encapsulamiento en cada problema sugerido, en el trabajo práctico intergradador o en el parcial.</p>	
<p>RA 9</p> <p>Programar algoritmos con estructuras de datos no lineales en memoria</p>	<p><b>Estructuras de Datos Lineales y No Lineales.</b> Estructuras lineales básicas: Tipo LIFO (pilas) y tipo FIFO (colas). Listas. Estructuras no lineales básicas: Árboles binarios de búsqueda (sin control de equilibrio) y Grafos</p>	<p><b>Estrategias:</b> Desarrollo de temas en forma de lección magistral participativa. Planteo de preguntas guía. Planteo de ejercicios y problemas que requieran definir, implementar y aplicar (en forma básica) estructuras de datos no lineales.</p>	<p><b>Instrumentos:</b> Cuestionarios planteados sobre los contenidos presentados. Problema de programación en modalidad de desafío: procesar un lote de datos concreto e informar los resultados.</p>	<p>Horas presenciales áulicas: 7 Horas teóricas: 4 Horas prácticas: 3 (Horas extra áulicas: 3)</p>

<p>temporal que puedan resolver diferentes problemas, utilizando árboles y grafos.</p>	<p>(implementación elemental en forma matricial). (Ver desglose y detalle de temas en el Ítem 10 – Programa Analítico [Unidad 8]).</p>	<p>Fomentar el aprendizaje colaborativo mediante el trabajo en grupos. Propuesta de estudio de casos amplios.</p> <p><b>Actividades:</b> Proponer respuestas a temas y preguntas guía que se planteen en el desarrollo de cada clase. Resolver los ejercicios y problemas, en forma individual o en forma grupal (según sea requerido). Proponer soluciones basadas en la aplicación elemental de estructuras de datos no lineales. Sugerir el planteo de consultas y dudas en los foros del curso, y la participación en la respuesta en esos mismos foros a esas consultas. Analizar variantes de solución para los casos amplios propuestos.</p>	<p>Problemas de complejidad incremental sugeridos en las guías de problemas propuestas en el curso. Trabajo práctico grupal integrador (resolución por etapas a lo largo del cursado). Examen Parcial a desarrollar en el laboratorio y en forma individual.</p> <p><b>Criterios Generales:</b> Participa en forma activa en el desarrollo de cada clase, aportando ideas claras y soluciones bien expresadas, pertinentes y entendibles tanto en expresiones coloquiales como en lenguaje y comunicación coloquial. Propone alternativas de solución para desafíos y problemas, mediante explicaciones adecuadas con justificación lógica y correctamente expresadas. Responde los cuestionarios en tiempo y forma.</p> <p><b>Criterios Específicos:</b></p>
--	--	---	---

			Implementa y utiliza correctamente (en forma básica) distintas estructuras de datos no lineales en diversas situaciones propias de problemas sugeridos, el trabajo práctico integrador y el parcial.	
<p>RA 10</p> <p>Distinguir las estrategias de resolución de problemas y técnicas básicas de análisis de algoritmos, para resolver en forma eficiente problemas diversos, considerando en forma amplia todos los recursos</p>	<p><b>Estrategias de Resolución de Problemas.</b> Clasificación básica. Fuerza bruta, recursividad, backtraking, divide y vencerás, algoritmos ávidos y programación dinámica.  (Ver desglose y detalle de temas en el Ítem 10 – Programa Analítico [Unidad 10]).</p>	<p><b>Estrategias:</b> Desarrollo de temas en forma de lección magistral participativa. Planteo de preguntas guía. Planteo de ejercicios y problemas que requieran seleccionar entre distintas estrategias de resolución de problemas. Fomentar el aprendizaje colaborativo mediante el trabajo en grupos. Propuesta de estudio de casos amplios.</p> <p><b>Actividades:</b> Proponer respuestas a temas y preguntas guía que se planteen en el desarrollo de cada clase. Resolver los ejercicios y problemas, en forma individual o en forma grupal (según sea requerido). Plantear soluciones genéricas mediante esquemas o respuestas</p>	<p><b>Instrumentos:</b> Cuestionarios planteados sobre los contenidos presentados. Problema de programación en modalidad de desafío: procesar un lote de datos concreto e informar los resultados. Problemas de complejidad incremental sugeridos en las guías de problemas propuestas en el curso. Trabajo práctico grupal integrador (resolución por etapas a lo largo del cursado). Examen Parcial a desarrollar en el laboratorio y en forma individual.</p> <p><b>Criterios Generales:</b> Participa en forma activa en el desarrollo de cada clase,</p>	<p>Horas presenciales áulicas: 7 Horas teóricas: 4 Horas prácticas: 3 (Horas extra áulicas: 3)</p>

<p>incorporados en la asignatura.</p>		<p>planteadas en papel o en documentos digitales. Plantear soluciones específicas en computadora mediante el lenguaje de programación propuesto. Proponer soluciones a diversos problemas, basadas en la aplicación elemental de distintas estrategias de planteo de algoritmos. Sugerir el planteo de consultas y dudas en los foros del curso, y la participación en la respuesta en esos mismos foros a esas consultas. Analizar variantes de solución para los casos amplios propuestos.</p>	<p>aportando ideas claras y soluciones bien expresadas, pertinentes y entendibles tanto en expresiones coloquiales como en lenguaje y comunicación coloquial. Propone alternativas de solución para desafíos y problemas, mediante explicaciones adecuadas con justificación lógica y correctamente expresadas. Responde los cuestionarios en tiempo y forma.</p> <p><b>Criterios Específicos:</b> Selecciona y aplica correctamente distintas estrategias de resolución de problemas, para los ejercicios sugeridos en guías.</p>	
<p>0</p>	<p>Indique los contenidos.</p>	<p>Actividades y estrategias de enseñanza para alcanzar el RA</p>	<p>Indicar criterios e instrumentos de evaluación.</p>	<p>Horas presenciales y horas extra áulicas. Desagregar en horas de teoría, práctica, laboratorio, etc..</p>

#### 14. Condiciones de aprobación

Se aplica un esquema de regularidad **común a todos los cursos**, y como resultado del mismo un estudiante puede quedar en alguna de las siguientes condiciones al finalizar el cursado:

Condición	Efecto
<b>Regular</b> = <b>Aprobación Directa</b> (aprueba la materia)	Un estudiante <b>Regular</b> es aquel que ha cumplido las exigencias mínimas del cursado. Y partir del ciclo 2024, todo estudiante que alcance la condición de <b>Regular</b> habrá alcanzado también la condición de <b>Aprobación Directa: habrá logrado la aprobación de la asignatura sin tener que rendir el examen final</b> . Debe inscribirse a examen final como si fuese a rendir, o esperar a que se genere en forma automática el acta con su nota.
<b>Libre</b> (recursa)	Si un estudiante hizo 4 o más de las 8 <i>evaluaciones principales</i> , pero no llegó a cumplir con las condiciones mínimas exigidas para la regularidad (= aprobación directa), se considera a ese estudiante como <b>Libre</b> . Será también <b>Libre</b> si no llegó a cumplir las condiciones de regularidad pero presentó el TPR el PR o el P5, sin importar si los aprobó o no. Debe recurrar la materia.
<b>Abandonó</b> (recursa)	Se considera que un estudiante <b>Abandonó</b> (y será registrado como tal) si quedó <b>Libre</b> pero haciendo menos de 4 de las 8 evaluaciones principales (y sin haber hecho el TPR ni el PR ni el P5). Debe recurrar la materia.
<b>No Cursó</b> (recursa)	Si el estudiante figura en los listados de Autogestión, pero nunca entró al Aula Virtual, o entró pero no completó ninguna de las evaluaciones previstas, se considera que ese estudiante <b>No Cursó</b> (y será registrado como tal). Debe recurrar la materia.

El esquema resumido de evaluaciones contempla las siguientes, en todos los cursos:

- 32 Cuestionarios teóricos. Cada uno admite dos intentos (el segundo cuenta como recuperatorio), y queda la nota promedio.
- 4 Desafíos de Programación. Cada uno admite 5 intentos (y esos intentos adicionales cuentan como recuperatorios), y queda la nota mayor.
- 4 Parciales (P1, P2, P3, P4).
- 1 Parcial Recuperatorio (PR) para recuperar uno y solo uno de los parciales.
- 4 Trabajos Prácticos (TP1, TP2, TP3, TP4), que son en realidad 4 etapas de un trabajo integrador.
- 1 Trabajo Práctico Recuperatorio (TPR) para recuperar uno y solo uno de los trabajos prácticos.
- 1 Parcial 5 especial (P5) a modo de coloquio final sobre temas de las últimas unidades. Admite dos intentos, y queda la nota mayor. El segundo intento, cuenta como recuperatorio.

Valen las siguientes observaciones:

- ✓ Las notas obtenidas en las evaluaciones citadas pertenecen a cuatro grupos, y a los efectos de determinar la condición final de cada estudiante se obtiene al final del cursado un promedio por cada grupo. Esos promedios se designan como PG1, PG2, PG3 y PG4 y se calculan como sigue (en todos los casos, se redondea al entero más próximo, y la fracción de 0.5 se redondea hacia arriba):

- **PG1:** promedio de notas entre los 32 cuestionarios y los 4 desafíos.
  - **PG2:** promedio de notas entre los cuatro trabajos prácticos (si el estudiante hizo el TPR, la nota del TPR reemplaza la nota de TP más baja a los efectos de este promedio).
  - **PG3:** promedio de notas entre los cuatro parciales (si el estudiante hizo el PR, la nota del PR reemplaza la nota de parcial más baja a los efectos de este promedio).
  - **PG4:** no es un promedio, sino directamente la nota final que se haya obtenido en el P5. Solo a los efectos de normalizar los nombres, se toma PG4 = P5.
- ✓ Se considera como evaluaciones principales a los cuatro trabajos prácticos (TP1, TP2, TP3 y TP4) y a los cuatro parciales (P1, P2, P3 y P4).
  - ✓ Los parciales se desarrollan en los laboratorios. La nota del PR reemplazará a la nota original más baja que el estudiante tenga en los cuatro parciales (o a un ausente). Estos parciales se evalúan en base a rúbricas (tablas de criterios) diseñadas por los docentes de la cátedra.
  - ✓ El P5 debe ser realizado por todos los estudiantes que estén en carrera para la regularización (= aprobación directa) de la materia. Se trata esencialmente de un cuestionario a resolver en el aula virtual a libro cerrado, estrictamente sobre temas que no llegan a evaluarse en los parciales ni en los trabajos prácticos (los últimos temas del programa de contenidos). El P5 admite dos intentos (el segundo cuenta como recuperatorio), y se registrará la nota más alta.
  - ✓ Los TP se desarrollan en grupos o en forma individual según sea indicado en los enunciados. El TPR puede hacerse en forma grupal o individual a decisión de los estudiantes. La nota del TPR reemplazará a la nota original más baja que el estudiante tenga en los cuatro trabajos prácticos (o a un ausente). Estos trabajos prácticos se evalúan en base a rúbricas diseñadas por los docentes de la cátedra.
  - ✓ Los Cuestionarios contienen preguntas de opciones múltiples. Se desarrollan en el aula virtual, a modo de refuerzo de lectura y estudio organizado y continuo de materiales. Pueden hacerse a libro abierto y desde sus casas, con tiempos de vencimiento entre una semana y dos semanas. Cada uno admite dos intentos (el segundo cuenta como recuperatorio), y quedará la nota promedio entre esos dos intentos.
  - ✓ Los Desafíos consisten en resolver algún problema y sólo informar los resultados. Se desarrollan en el aula virtual, a modo de experiencia de programación en contexto de exigencia, que requiera pruebas y revisiones hasta cumplir los requerimientos propuestos. Pueden hacerse a libro abierto y desde sus casas, con tiempos de vencimiento entre dos semanas y tres semanas. Cada uno admite hasta cinco intentos (cada uno de los adicionales cuenta como recuperatorio), y quedará la nota más alta entre esos cinco intentos.
  - ✓ Esencialmente, se aplica un esquema de evaluación continua, con reglas de trabajo firmes pero claras. En todo momento los estudiantes saben cuál será el esquema de evaluaciones, cómo serán evaluados, y cuándo les serán entregadas sus devoluciones y calificaciones.
  - ✓ El aula virtual es única y centralizada para todos los cursos, con exactamente el mismo y único modelo de intervención para todos los cursos, lo que facilita el proceso de avance y de control, la integración entre los estudiantes, y promueve un esquema de estricta justicia durante el cursado.

Con todo lo anterior, la condición final de cada alumno surgirá entonces del siguiente esquema:

⇒ **Condición: Regular (= Aprobación Directa):**

El estudiante debe cumplir **todas** las siguientes condiciones al finalizar el cursado:

- a. Promedio del Grupo 1 (PG1) de Evaluaciones (Cuestionarios y Desafíos) mayor o igual a 6 (es decir:  $PG1 \geq 6$ ).
- b. Promedio del Grupo 2 (PG2) de Evaluaciones (Trabajos Prácticos) mayor o igual a 6 (es decir:  $PG2 \geq 6$ ). Es obligatorio hacer y aprobar el TP4 (con nota mayor o igual a 4(cuatro)), pero no es obligatorio aprobar los otros tres, mientras el PG2 sea mayor o igual a 6. Aplica el TPR si es necesario para levantar el promedio (ya sea para aprobar el TP4 como para levantar el promedio).
- c. Promedio del Grupo 3 (PG3) de Evaluaciones (Parciales) mayor o igual a 6 (es decir:  $PG3 \geq 6$ ). Es obligatorio hacer y aprobar el P4 (con nota mayor o igual a 4(cuatro)), pero no es obligatorio aprobar los otros tres, mientras el PG3 sea mayor o igual a 6. Aplica el PR si es necesario (ya sea para aprobar el P4 como para levantar el promedio).
- d. Parcial 5 (equivalente en sí mismo al PG4) mayor o igual a 6 (es decir:  $P5 (= PG4) \geq 6$ ). Recordar que se cuenta con dos intentos si fuese necesario y que vale la más alta de las notas.

⇒ **Condición: Libre:**

- a. Si un estudiante no llegó a la condición mínima de *regular (= aprobación directa)* por no haber cumplido con una o más de las cuatro condiciones para ser regular, pero hizo 4 o más de las 8 evaluaciones principales o presentó el TPR o el PR o el P5 (sin importar si los aprobó o no), entonces quedará en condición de **Libre**. Deberá recurrar la materia.

⇒ **Condición: Abandonó:**

- a. Se considera que un estudiante **Abandonó** (y será registrado como tal) si quedó *libre* pero haciendo menos de 4 de las 8 evaluaciones principales (y sin haber hecho el TPR ni el PR ni el P5). Deberá recurrar la materia.

⇒ **Condición: No cursó:**

- a. Se considera que un estudiante No Cursó (y será registrado como tal) si figura en los listados de Autogestión, pero nunca entró al Aula Virtual, o entró pero no completó ninguna de las evaluaciones previstas. Deberá recurrar la materia.

En general, para el cálculo de las notas de parciales y trabajos prácticos se aplicará una escala de valoración y conversión de puntajes que tendrá en cuenta una serie de ítems que oportunamente serán indicados a los alumnos. Cada ítem equivaldrá a un cierto porcentaje máximo de completitud y buen desarrollo, y de acuerdo con la suma total de porcentajes se calculará la nota final en cada evaluación.

Solo a efectos informativos, indicamos que las escalas de notas que se aplican son dos: una para el proceso de regularidad (durante el cursado) y la otra para el proceso de examen final (que ya no aplica en AED para estudiantes a partir del ciclo 2024, debido a que todo estudiante obtiene la condición de Aprobación Directa si logra regularizar, pero sí aplica para estudiantes de ciclos anteriores que todavía deben hacer el examen final). Las dos tablas de notas son las que siguen (notar que, durante el cursado, la nota mínima de aprobación es 4(cuatro), pero en un examen final la nota mínima para aprobar es 6(seis)):

**Escala de notas para el proceso de *Regularidad* (a aplicar DURANTE el cursado)**

NOTAS	PORCENTAJE	CALIFICACIÓN
1		No Aprobado
2		No Aprobado
3		No Aprobado
4	55% a 57%	Aprobado
5	58% a 59%	Aprobado
6	60% a 68%	Aprobado
7	69% a 77%	Aprobado
8	78% a 86%	Aprobado
9	87% a 95%	Aprobado
10	96% a 100%	Aprobado

**Escala de Notas para *Examen Final***

NOTA	PORCENTAJE	CALIFICACIÓN
1		Insuficiente
2		Insuficiente
3		Insuficiente
4		Insuficiente
5		Insuficiente
6	60% a 68%	Aprobado
7	69% a 77%	Bueno
8	78% a 86%	Muy Bueno
9	87% a 95%	Distinguido
10	96% a 100%	Sobresaliente

**15. Modalidad de examen**

Debido a que a partir del ciclo 2024 los estudiantes que regularicen logran también el estado de aprobación directa, no habrá entonces examen final para estudiantes que hayan regularizado a partir de 2024, pero obviamente, los exámenes finales continuarán en forma normal para estudiantes que hayan regularizado antes de 2024.

La modalidad de examen final para quienes deban rendirlo por haber regularizado antes de 2024, consiste en lo siguiente:

- Un cuestionario teórico a desarrollar en el aula del examen, similar al P5 del cursado regular, que debe aprobarse con 6 o más.
- Un práctico que consiste en el desarrollo de un programa en base a un enunciado provisto en ese momento, que debe ser aprobado con 6 o más.
- Si alguna de las dos instancias anteriores es reprobada, el examen final termina y el estudiante reprobaba.

- Si ambas fueron aprobadas, la nota final será el promedio redondeado al entero más próximo entre ambas notas.

Los estudiantes a partir del ciclo 2024 sólo deben inscribirse como si fuesen a rendir y en la medida de lo posible pasar al aula del examen el día en que se sustancie, para registrar su nota final en el acta del examen y en su libreta. O esperar a que se genere en forma automática el acta de ese examen. La firma de la libreta con la nota del examen final puede hacerse también en cualquier momento posterior a la fecha oficial del examen (siempre que el estudiante tenga acceso a Autogestión 4 para convalidar esa nota).

### 16. Recursos necesarios

#### **Espacios Físicos (aulas, laboratorios, equipamiento informático, etc.).**

Se requiere contar con aulas comunes con capacidad para receptor a la totalidad de los alumnos inscriptos, y tales que cuenten además con ventilación, iluminación adecuada, conectividad wifi, enchufes para la alimentación de notebooks y posibilidad de usar cañones de proyección. En general, las clases teóricas se desarrollan en este tipo aulas (al menos en el primer semestre). Para el desarrollo de las clases prácticas se espera contar con aulas en laboratorios de informática, con capacidad para la totalidad de los alumnos inscriptos, y que cuenten con equipamiento en el que ya esté instalado el software requerido para la asignatura. No se descarta que en algún momento también las clases teóricas se desarrollen en los laboratorios.

#### **Recursos tecnológicos de apoyo (proyector multimedia, software, equipo de sonido, aulas virtuales, etc.).**

La cátedra de AED cuenta con un Aula Virtual centralizada y muy completa y funcional, montada sobre la plataforma Moodle de la FRC, que contiene todos los materiales de estudio, evaluaciones y medios de comunicación. Por lo tanto, se espera contar con la plataforma Moodle disponible y con soporte técnico cuando sea requerido.