

**INSTITUTO SUPERIOR  
DEL PROFESORADO  
“Dr. Joaquín V. González”  
Profesorado en  
Informática**

**Año 2022**

# Programación III

Prof. Silvia Pilar Rodríguez

---

**2022***Año del 40° Aniversario de la Guerra de Malvinas**En homenaje a los veteranos y caídos en defensa de las Islas Malvinas y el Atlántico Sur***Nivel:** Superior**Carrera:** Profesorado en Informática**Eje:** Formación general**Instancia curricular:** Programación III**Cursada:** Anual**Carga horaria:** 5 horas cátedra semanales**Profesora:** Silvia Pilar Rodríguez**Año:** 2022**Fundamentación:**

La actualización constante y la capacitación permanente son características indispensables para el ejercicio de la profesión docente. Esta afirmación, avalada desde varias perspectivas, es aplicable a cualquier área de conocimiento.

Sin embargo, en la cuestión informática, el carácter dinámico de la disciplina en sí misma hace que dicha actualización sea uno de los ejes troncales de trabajo. Los avances permanentes en el área, más allá del trabajo docente y de las distintas miradas pedagógicas sobre la enseñanza de la informática, requieren del profesor una disposición a la investigación y a la búsqueda de nuevas alternativas que es mucho más demandante que en otras disciplinas.

Ahora bien, paradójicamente esta misma condición de constante dinamismo, hace muy difícil la tarea de anticipar cuáles son los contenidos específicos que deben dictarse en una materia de este tenor, vale decir, qué lenguaje específicamente trabajar en este espacio.

A lo largo de la muy reciente historia de la programación, distintos paradigmas se han instalado según la época y algunos se han popularizado más que otros. No obstante, subsiste un eje común a la mayoría de ellos, aun cuando sobre esta base se hayan elaborado notables modificaciones que en ocasiones parecen haberla “enterrado”: las estructuras de programación y de datos.

Más allá de cualquier lenguaje de codificación de alto nivel y de cualquier técnica algorítmica, conocer estas estructuras básicas hace a la esencia del buen programador, que no es solamente el que logra codificar un programa que funcione, sino fundamentalmente elaborar un diseño cuyo resultado sea un programa estable, eficiente, comprensible y reutilizable. Estas

características, inherentes a la programación, se hacen mucho más necesarias en un profesional que no va a dedicarse a brindar prontas soluciones a un usuario, sino que va a ser el responsable del aprendizaje de los futuros programadores.

No hablamos en este espacio de codificación en un lenguaje determinado, si bien como es natural un compilador específico se hace indispensable. Más bien apuntamos a técnicas de programación modular orientada a objetos, con manejo de estructuras estáticas y dinámicas según la situación, y al discernimiento acerca de la mejor opción para la obtención de los resultados buscados. Por otra parte, una de las capacidades que el futuro docente debe desarrollar, es la autonomía en la elección de dispositivos que faciliten la enseñanza de los contenidos, para lo cual deberá conocer alternativas, ventajas y desventajas de cada una, y generar sus propuestas de trabajo alrededor de estas consideraciones. Es por esto que la cátedra propone un lenguaje pero se aviene a las sugerencias e inquietudes de los futuros profesores, siempre que estas se encuentren enmarcadas en el presente plan de trabajo.

En principio, y para poder sostener concretamente la metodología de programación estructurada, se utilizará el lenguaje Python por tratarse de un lenguaje versátil, adaptable a múltiples plataformas y muy pertinente para la programación de sistemas remotos. De este modo, la codificación en este lenguaje será una herramienta para la aplicación de la metodología de trabajo propuesta por la cátedra y no en un contenido en sí mismo.

### **Objetivos**

Que el futuro docente:

- Conozca y pueda comparar ventajas y desventajas de los distintos lenguajes del paradigma de la programación orientada a objetos.
- Sea capaz de seleccionar distintas alternativas metodológicas según la situación a la que se enfrenta.
- Apunte a la correcta escritura de un programa escribiendo un código ejecutable correcto, estable y eficiente, y diseñando un código fuente comprensible y reutilizable.
- Adquiera conocimientos que le permitan indagar en otras alternativas de codificación.
- Desarrolle la capacidad de evaluar, co-evaluar y auto-evaluar los resultados ejecutables de los programas y sus códigos fuente.
- Valore los lenguajes de programación como una herramienta potente a la hora de diseñar software educativo.
- Sea capaz de interpelarse a sí mismo en su tarea a partir de una práctica reflexiva

- Entienda su propia tarea docente como un espacio de investigación y producción de conocimiento
- Diseñe un proyecto concreto de trabajo áulico que utilice estrategias didácticas mediadas por TIC.

## **Contenidos**

### **EJE CONCEPTUAL: Los paradigmas de programación y la programación orientada a objetos**

*Objetivo: Conocer las ventajas y desventajas de los distintos paradigmas de programación y sus posibilidades de aplicación.*

*Aplicación: Investigación bibliográfica y en la web.*

- La programación estructurada. Fundamentos. El teorema de Dijkstra.
- Otros paradigmas de programación: orientada a objetos, orientada a eventos, por listas, lógica, lineal, etc. Ventajas, desventajas y aplicaciones de cada uno de ellos.
- OOP: Herencia, encapsulación y polimorfismo
- Clases y objeto. Propiedades de una clase
- Métodos. Procedimientos y funciones
- Pasaje de parámetros: por valor y por referencia. Variables locales, privadas y públicas

### **EJE METODOLÓGICO: Las estructuras de programación**

*Objetivo: Retomar y transferir los conocimientos adquiridos en Programación I y II.*

*Aplicación: Algoritmos.*

- Escritura y organización de un programa.
- Entrada y salida de datos. Interfaces y librerías.
- Variables. Declaraciones. Asignaciones.
- Tipos de datos.
- Vectores. Matrices. Registros. Archivos.
- Operadores aritméticos y lógicos.
- La programación modular
- Alcance de las variables. Ámbito. Variables locales y globales. Parámetros.
- Funciones recursivas
- Estructuras de asignación dinámica.

### **EJE DE APLICACIÓN PRÁCTICA: Programación para la gestión de Datos**

*Objetivo: Recuperar los conceptos fundamentales de la tecnología de Bases de Datos y aplicarlos a la programación de sistemas locales y remotos*

*Aplicación: Sistemas de Gestión de Bases de Datos*

- Modelos de datos

- Ventajas del enfoque de Bases de Datos. Comparación con los archivos tradicionales
- Sistema de Gestión de Bases de Datos (DBMS)
- Modelo Entidad – Relación. Reducción a tablas.
- Modelo Relacional. Dependencias funcionales e integridad referencial.
- Normalización.
- Lenguaje SQL. Consultas
- ABM de datos
- Funciones de librería: conversión, entrada y salida de datos, manejo de archivos.
- Funciones creadas por el usuario.
- Aplicaciones móviles.
- Arquitectura cliente-servidor. Protocolos de comunicación.
- Scripts.
- Tratamiento de datos remotos. Colisiones.
- Gestión de eventos en la Web. Seguridad y consistencia.

### **EJE PEDAGÓGICO-DIDÁCTICO: Trabajo de integración para un proyecto áulico**

**Objetivo:** *Desarrollar un sistema de software relativo a la educación y una aplicación para dispositivos móviles*

**Aplicación:** *Programación de sistemas*

- Creación de un proyecto propio desde la etapa de análisis, confección del diseño y codificación de la programación, y fundamentación de las decisiones, que involucre los contenidos abordados y se oriente a las necesidades pedagógicas de los alumnos a quienes está dirigido.

### **Modalidad de trabajo/Estrategias Didácticas:**

La modalidad de trabajo en la materia es de carácter teórico, de análisis, y producción práctica. Esto implica que a lo largo del año se abordarán los distintos ejes propuestos desde las diversas metodologías, con espacios de intercambio y de consulta de manera de brindar al alumno el acompañamiento necesario para abordar convenientemente los temas.

Para la elaboración del trabajo final, se trabajará con la modalidad ABP a partir del proyecto presentado por los alumnos que será el cierre de la materia.

En cualquier tipo de metodología de trabajo, la participación activa del estudiante y su protagonismo en la relación que establece con el conocimiento es fundamental. Asimismo, la socialización de las prácticas y de los proyectos constituye una importante instancia de aprendizaje para la adquisición de los conceptos.

Como recurso fundamental, el alumno deberá contar con su computadora conectada a Internet, que le permita concretar los proyectos y acceder a la información que se pondrá a

disposición en el Campus. También se recurrirá a apuntes, videos tutoriales y presentaciones visuales con los contenidos conceptuales y los procedimientos que el alumno deberá incorporar.

### **Trabajos prácticos, instrumentos y criterios de evaluación**

Tanto los procesos de enseñanza y de aprendizaje en la materia como los contenidos, se encuentran interrelacionados, por lo que cada trabajo práctico resulta integrador de todos los conceptos anteriores.

El alumno dará cuenta de los conceptos y procedimientos adquiridos a través de tres instrumentos a lo largo del año:

- Trabajos prácticos sobre cada tema.
- Dos instancias evaluativas parciales individuales.
- Un trabajo grupal con carácter integrador correspondiente al eje de Aplicación Práctica. Para esta última instancia es indispensable la presentación de una bitácora de recorrido por los trabajos prácticos parciales, que asegure el seguimiento permanente de los temas trabajados en clase.

### **Evaluación, aprobación y acreditación de las instancias curriculares**

#### **Con examen final:**

- a) Asistencia: cumplimentar el 60% de las participaciones propuestas en el aula virtual.
- b) Trabajos prácticos: tener aprobados el 70%.
- c) Instancias parciales: para aprobar cada uno de ellos se requerirá una calificación mínima de 4 puntos sobre 10. Cada evaluación parcial tendrá un recuperatorio. Cuando se presenta esta situación, se considerará, a los efectos del promedio, solamente la nota del recuperatorio.
- d) Trabajo final integrador: deberá entregarse un avance del trabajo según las consignas del mismo.

**Sin examen final:** (solo para talleres, seminarios o cursos que comienzan con un máximo de 25 inscriptos)

- a) Las correlatividades previas de la instancia curricular, deben estar aprobadas al mes de mayo del año en que se cursa la asignatura por promoción sin examen final. En caso de que en el mes de mayo el alumno no apruebe las correlativas anteriores o no las rindan, pasará automáticamente al régimen de promoción con examen final.
- b) Asistencia: cumplimentar el 80% de las participaciones propuestas en el aula virtual.
- c) Trabajos prácticos: tener aprobados el 100%.
- d) Instancias parciales: para aprobar cada uno de ellos se requerirá una calificación mínima de 7 puntos sobre 10. Cada evaluación parcial tendrá un recuperatorio. Cuando se presenta esta situación, se considerará, a los efectos del promedio, solamente la nota del recuperatorio.

- e) Trabajo final integrador: deberá entregarse el trabajo terminado según las consignas del mismo.

**Criterios de evaluación final de la instancia curricular:**

En todos los casos se rendirán frente a un tribunal de profesores, ya sea en modalidad virtual o presencial.

**Para el alumno inscripto en condición de regular:**

1. Promoción con examen final (según especificaciones dadas)
2. Promoción sin examen final (según especificaciones dadas)

**Para el alumno inscripto en condición de libre:**

Evaluación teórico-práctica, escrita y en máquina de todos los contenidos, con la resolución de un problema integral.

*Los exámenes libres serán indefectiblemente escritos y orales y abarcará el programa completo del curso con la bibliografía indicada. El examen escrito es eliminatorio y quedará archivado,*  
**Resolución del Consejo Directivo oct/2013**

**Bibliografía específica obligatoria:**

Por tratarse de un abordaje de diseño práctico y taller, se trabajará con apuntes de cátedra, sitios web de consulta online y tutoriales sobre programación.

**Firma del Profesor**