



<b>SEMINARIO DE LENGUAJES (Opción "Python")</b>	<b>Carrera:</b> <i>Licenciatura en Informática</i> Plan 2015 <i>Licenciatura en Sistemas</i> Plan 2015 <i>Analista Programador Universitario</i> Plan 2015 <i>Analista en TIC</i> Plan 2017
	<b>Año:</b> 2° <b>Carácter:</b> Electiva
	<b>Año 2018</b> <b>Duración:</b> Semestral
	<b>Correlatividad:</b> Taller de Programación <b>Profesor:</b> Claudia Banchoff, Viviana Harari <b>Hs. semanales:</b> 6 hs.

## FUNDAMENTACIÓN

Este seminario está orientado a que el alumno ponga en práctica los conceptos vistos en primer año, enfatizado el trabajo sobre la computadora. Los alumnos reforzarán estos conceptos y aprenderán cómo se los implementa en un lenguaje distinto al utilizado hasta este momento.

## OBJETIVOS GENERALES

Desarrollar una aplicación concreta, profundizando los conocimientos obtenidos en los primeros cursos vinculados con algoritmos y programación, permitiendo al alumno desarrollar un estudio teórico-práctico de algún lenguaje de programación (el lenguaje puede variar con los cambios tecnológicos), poniendo énfasis en el análisis formal de las características del lenguaje y su comparación con los que el alumno conociera a ese momento.

## CONTENIDOS MÍNIMOS

Estudio de un lenguaje de programación en el que se desarrollen aplicaciones concretas. En lo posible la oferta de lenguajes será variable y actualizada con el cambio tecnológico.

## PROGRAMA ANALÍTICO

Unidad I. Conceptos de software libre. Un poco de historia. El proceso de ejecución de un programa escrito en Python. Características generales.

Unidad II. Sintaxis básica del lenguaje. Tipos predefinidos. Declaración de variables. Conversiones. Estructuras de control. El formato de un programa de Python. Definición de funciones y módulos. Pasaje de parámetros. Alcances.

Unidad III. Estructuras de datos. Listas, tuplas, conjuntos y diccionarios. Manejo de archivos.



Unidad IV. Abstracción de datos en Python. Conceptos básicos de programación orientada a objetos. Manejo de excepciones.

Unidad V. Resolución de problemas sencillos. Comparación de soluciones a problemas ya vistos destacando las diferencias con los lenguajes de programación conocidos hasta el momento.

Unidad VI. Librerías externas. Generación de gráficos. Procesamiento de formatos típicos como JSON y CVS. Análisis de textos.

Unidad VII. Programación de videojuegos. Motores de juegos para Python. Características generales. Módulos. Manejo de objetos y eventos básicos. Diseño de un juego sencillo.

## METODOLOGÍA DE ENSEÑANZA

La asignatura es de tipo taller, la teoría y práctica se encuentran estrechamente vinculadas. Se trabaja sobre los conceptos aprendidos en la única asignatura de programación que los alumnos tuvieron hasta esta instancia, comparando y fortaleciendo dichos conceptos pero encarados desde la práctica con lenguaje Python.

Los teóricos son explicaciones semanales donde se desarrolla, en forma conceptual, cada concepto a trabajar. Los mismos son luego tomados en las instancias prácticas.

Se plantea un desarrollo final donde se integran todos los conceptos aprendidos. Este trabajo final incluye el software correspondiente y un informe. Generalmente se trabaja en el desarrollo de juegos educativos para el nivel primario e inicial, que presentan una complejidad sencilla y que pueden ser abordados por un alumno de segundo año.

En los horarios de práctica, se trabaja sobre la resolución de los ejercicios propuestos, en donde los docentes realizan la tarea de supervisión, atendiendo las dificultades planteadas y revisando la utilización de los lineamientos conceptuales.

Se trabaja con los siguientes recursos:

- Guías de orientación para los trabajos de producción.
- Diapositivas, videos, libros y tutoriales.
- Proyector, computadora, demostraciones de usos de herramientas con ejemplos en vivo.
- EVEA de soporte: <https://catedras.info.unlp.edu.ar/>

## EVALUACIÓN

A modo de ejercitación y evaluación se plantean, a lo largo de la cursada, entregas de ejercicios que los alumnos deben desarrollar y entregar en las prácticas y teorías.

Como evaluación final se plantea un trabajo integrador con dos entregas obligatorias. Las mismas deben ser defendidas por los alumnos en forma de coloquio, contestando preguntas relacionadas con los conceptos teóricos involucrados, requisito para la aprobación de la cursada. Estas entregas son de seguimiento y de evaluación con calificación.

Al finalizar la cursada existe una instancia de evaluación donde el alumno expone, en forma completa, el trabajo realizado.



La materia se aprueba con el 75% de los ejercicios prácticos y teóricos aprobados y con el trabajo integrador aprobado.

Se realiza una encuesta sobre los conocimientos iniciales de los alumnos en la que se releva, además, otra información de interés como ser su situación laboral (evaluación diagnóstica).

### BIBLIOGRAFÍA

- Python Programming: An Introduction to Computer Science. John M. Zelle
- Introduction to Computing and Programming in Python, A Multimedia Approach. Mark Guzdial.
- Python para todos. Raúl González Duque.
- Beginning Python: From Novice to Professional - Magnus Lie Hetland.
- An Introduction to Python. Guido van Rossum.
- Learning Python. O'Reilly.

### INTEGRACIÓN CON OTRAS ÁREAS

Los trabajos realizados se han planteado pensando en su integración a dos proyectos de extensión e investigación que trabajan con software libre y escuelas.

Este tipo de desarrollo da una motivación particular a los alumnos, dado que pueden evaluar sus propias aplicaciones con usuarios finales reales en una muestra organizada por la cátedra.

Se motiva a los alumnos para que se sumen a los proyectos. Otros alumnos demostraron interés en participar como colaboradores alumnos de la materia.

### CRONOGRAMA DE CLASES Y EVALUACIONES

Semana del 13 de marzo: encuesta Inicial y presentación de la materia.
Clase 1: conceptos básicos. Tipos de datos.
Explicación de Práctica Inicial: explicación sobre el uso alguno de los IDEs propuestos y las pautas para realización de las prácticas. Taller de git.
Clases 2 y 3: estructuras de control y tipo de datos avanzados.
Semana del 10 de abril: primera evaluación teórica
Semana del 17 de abril: primera evaluación práctica
Clase 4: módulos y funciones.



**UNIVERSIDAD NACIONAL DE LA PLATA**  
**FACULTAD DE INFORMÁTICA**

---

Clase 5 y 6: manejo de archivos.
Semana del 8 de mayo: segunda evaluación teórica
Semana del 15 de mayo: segunda evaluación práctica
Clase 7: presentación de la API a utilizar.
Clases 8 y 9: introducción a la programación orientada a objetos. Manejo de excepciones.
Semana del 29 de mayo: tercera evaluación teórica.
Semana del 5 de junio: segunda evaluación práctica
Clases 10, 11 y 12: explicación detallada de la API a utilizar y de las librerías necesarias.
Semana de 19 de junio: primera instancia de recuperación de evaluaciones.
Semana del 19 de junio: entrega trabajo integrador.
Semana de 3 de julio: segunda instancia de recuperación de evaluaciones.
Semana del 30 de julio: re-entrega del trabajo final.

**Contacto de la cátedra**

Mail: [python@info.unlp.edu.ar](mailto:python@info.unlp.edu.ar)

Plataforma virtual: [catedras.info.unlp.edu.ar](http://catedras.info.unlp.edu.ar)