

1. Información del equipo pedagógico y horarios de atención a estudiantes

Sección 1

Profesor: José Sebastián Ñungo Manrique (js.nungo@uniandes.edu.co)

Clase: Miércoles 5:00 pm – 6:20 pm

Horario de atención: Cita previa por correo ([virtual](#))

Sección 2

Profesora: Catalina María Bernal Murcia (cm.bernal10@uniandes.edu.co)

Clase: Martes 5:00 pm – 6:20 pm

Horario de atención: Cita previa por correo (virtual)

2. Descripción del curso

El curso Taller de Python busca que los y las estudiantes adquieran una comprensión general del lenguaje de programación Python y su utilidad para el análisis de datos, la investigación económica y otras aplicaciones académicas y profesionales. El énfasis del curso prioriza la importación, organización, manejo de datos y la presentación de resultados a través de herramientas que permitan una comunicación efectiva. Además, busca desarrollar pensamiento lógico y computacional para la solución de problemas, la toma de decisiones y el discernimiento ético en los procesos de analítica en la actualidad.

Este curso no profundiza en conceptos estadísticos o econométricos; en cambio, busca que los y las estudiantes entiendan la estructura general de programación en Python y la complementen con los conocimientos adquiridos en otras clases y el apoyo de material adicional. Al final, se espera que los y las estudiantes se sientan cómodo/as utilizando Python para resolver problemas relacionados con análisis de datos. Asimismo, se espera que logren identificar las diferentes aplicaciones de dicho lenguaje de programación y puedan continuar utilizándolo tanto en actividades académicas como laborales.

3. Resultados de aprendizaje

- Familiarizar a los estudiantes con el lenguaje de programación Python.
- Motivar el uso de Python en diferentes contextos mediante la presentación de casos prácticos.
- Proporcionar herramientas para el manejo y uso de bases de datos en Python.
- Mostrar a los estudiantes paquetes y comandos frecuentemente usados en Python.
- Incentivar el trabajo en equipo para la solución de problemas.
- Enseñar a los estudiantes formas de automatizar tareas en Python.
- Proporcionar herramientas para que los estudiantes puedan continuar de manera autónoma su aprendizaje de paquetes y aplicaciones de Python.

- Desarrollar el pensamiento lógico y computacional para la toma de decisiones y solución de problemas.
- Incentivar el pensamiento crítico frente a dilemas éticos en el uso de tecnologías computacionales.

Al finalizar el curso, los y las estudiantes podrán:

- Identificar el contenido y la utilidad de los paquetes de Python ante diferentes problemas o escenarios.
- Localizar los archivos de ayuda sobre cada paquete de Python.
- Construir sus propias funciones para resolver problemas e importar y limpiar datos.
- Explicar, de forma escrita, el contenido de sus propias funciones para describir la lógica de programación que constituye su solución de forma que cualquier persona pueda entenderlo.
- Manipular datos para obtener información relevante de acuerdo con distintos escenarios.
- Asociar funciones específicas de Python para responder una pregunta o resolver un problema.
- Identificar diferentes tipos de datos, su estructura y atributos en Python.

4. Cronograma

Semana 1 (8 – 11 de agosto)

Clase 1: Introducción a Python

- ¿Qué es? ¿Para qué?
- Funciones internas de Python.
- Operaciones matemáticas.

Semana 2 (14 – 18 de agosto)

Clase 2: Introducción a Python II

- Estructuras de datos básicas: int, str, float, listas, diccionarios, booleanos.
- Funciones de identificación y conversión.
- Asignar objetos, mutarlos y borrarlos.
- Métodos y atributos.

Semana 3 (22– 25 de agosto)

Clase 3: Exploración de objetos y condicionales.

- Indexación de listas y diccionarios.
- Condicionales.
- If, elif y else.

Semana 4 (28 de agosto – 1 de septiembre)

Clase 4: Módulos y Funciones I

- Uso de funciones.
- Definición de funciones.
- Instalación e importación de paquetes.
- Numpy.

Semana 5 (4 – 8 de septiembre)

Clase 5: Funciones II

- Funciones recursivas.
- Algoritmos.

Semana 6 (11 – 15 de septiembre)

Clase 6: Más condicionales y Loops

- Controles.
- For y while.
- Map.
- Zip.

Semana 7 (18 – 22 de septiembre)

Clase 7: TALLER 1

Semana 8 (25 – 29 de septiembre)

Clase 8: Bases de datos I

- Pandas
- Importación y exportación de datos
- Métodos de los DataFrame y Series
- Manipulación de bases de datos: indexación, operaciones y agrupación.

Semana 9 (2 – 6 de octubre)

Semana de Receso

Semana 10 (9 – 13 de octubre)

Clase 9: Bases de datos II

- Manipulación de bases de datos: muestreos y agrupaciones.
- Unión: merge, join, concat.
- Estadísticas descriptivas.

Semana 11 (17 – 20 de octubre)

Clase 10: Bases de datos III

- Continuación de manipulación de bases de datos: muestreos y agrupaciones.
- Unión: merge, join, concat.
- Continuación de estadísticas descriptivas.

Semana 12 (23 – 27 de octubre)

Clase 11: Visualización de datos

- Matplotlib.
- Seaborn.

Semana 13 (30 de octubre – 3 de noviembre)

Clase 12: Repaso de Pandas y Matplotlib

- Uso de funciones vistas previamente.
- ¡No todo tiene una única forma de resolverse!

Semana 14 (7 – 10 de noviembre)

Clase 13: TALLER 2

Semana 15 (14 – 17 de noviembre)

Clase 14: Econometría e Introducción a Machine Learning

- Statmodels.

- Scikit-learn.
- Modelos discretos.
- Aprendizaje de máquinas no supervisado.

Semana 16 (20 – 24 de noviembre)

Clase 15: Introducción a Machine Learning II y a la Ética Algorítmica

- Aprendizaje de máquinas supervisado.
- Dilemas éticos en el uso y procesamiento de datos.
- Herramientas de auditoría algorítmica: ¿por qué es importante?

Semana 17 (27 de noviembre – 1 de diciembre)

Clase 16: TALLER 3

5. Referencias

Las referencias serán mencionadas en el material de clase, sujetas a la necesidad del contenido.

6. Metodología

Este curso introduce a los y las estudiantes al uso de Python y presenta algunas aplicaciones que pretenden motivar su continuo aprendizaje y utilización durante el curso e incluso, después de haberlo terminado. Este curso se realizará de manera presencial y cada clase estará dividida en dos partes: durante la primera parte el profesor introducirá técnicas, comandos y conceptos relacionados con el uso de Python y utilizará ejemplos que lo ilustran; durante la segunda parte los estudiantes deberán trabajar individualmente (con apoyo de sus compañero/as) en un ejercicio el cual deberán entregar al finalizar la clase.

En el semestre se realizarán tres (3) talleres en clase sobre los temas que se abordan. Estos talleres deben ser desarrollados por los estudiantes durante la clase, sin embargo, tendrán hasta la medianoche del domingo de esa semana para entregarlos (esta fecha de entrega puede estar sujeta a modificaciones). Adicionalmente, durante el semestre se evaluará la participación de los estudiantes a través de participaciones en clase y por fuera, ya sea a través de preguntas y, mediante la colaboración con sus colegas. Además, el porcentaje de participación incluye la calificación de **cuatro** ejercicios de clase, seleccionados de manera aleatoria al finalizar las clases (semana 17), que serán evaluados.

Es recomendable mas no obligatorio realizar una preparación previa del contenido de la clase sujeto al programa anteriormente descrito.

7. Evaluaciones

Los talleres se evaluarán teniendo en cuenta:

- 1) Organización del código (estructura y comentarios),
- 2) Que el código genere un resultado correcto o la lógica de la solución sea coherente y,

3) Número de errores en el código.

Los talleres deben ser presentados a través de Bloque Neón.

Talleres: 90% (3x30% c/u)

Participación/Ejercicios: 10%

- ✓ Para aprobar el curso cada estudiante debe alcanzar una nota igual o superior a 3.0.
- ✓ La calificación definitiva será numérica de 1.5 a 5.0, en unidades y décimas.
- ✓ Reclamos: Según los artículos 64, 65 y 66 del [Reglamento General de Estudiantes de Pregrado](#), el estudiante tendrá **cuatro** días hábiles después de la entrega de la evaluación calificada para presentar un reclamo. **El profesor magistral** responderá al reclamo en los **cinco** días hábiles siguientes. Si el estudiante considera que la respuesta no concuerda con los criterios de evaluación podrá solicitar un segundo calificador al Consejo de la Facultad en los **cuatro** días hábiles posteriores a la recepción de la decisión del profesor.
- ✓ Fraude académico: artículo 116 del [Reglamento General de Estudiantes de Pregrado](#). En caso de Fraude los estudiantes involucrados tendrán 0 en la actividad correspondiente.

8. Asistencia

Según los artículos 43 y 44 del [Reglamento General de Estudiantes de Pregrado \(RGEP\)](#), los estudiantes pueden no asistir hasta al 20% de las sesiones. En caso de superar este porcentaje, se define una consecuencia sobre el curso. Sin embargo, en este curso no se hará control de la asistencia y, por ende, no habrá una penalización por incumplir los artículos mencionados anteriormente. Se recomienda que, en caso de no poder asistir a la sección en la que se encuentra inscrito, el estudiante asista a la otra sección.

9. Políticas generales de los cursos de Economía y fechas importantes

Los y las estudiantes deben consultar [este enlace](#), donde se encuentran las reglas sobre asistencia a clase, excusas válidas, fraude académico y faltas disciplinarias, reclamos, políticas de bienestar y fechas importantes del semestre.