

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

**Profesor:** Fayber Acosta Pardo ([fa.acosta1@uniandes.edu.co](mailto:fa.acosta1@uniandes.edu.co))

**Clase:** Martes 6:30 a.m. – 7:50 a.m. - Salón SD\_301

**Horario de atención:** Cita previa por correo (virtual)

## **2. Descripción del curso**

En las últimas décadas el manejo de softwares estadísticos se ha convertido en una habilidad indispensable para el análisis de datos y para automatizar tareas cotidianas. 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.

Debido a la condición de taller del curso es imprescindible el compromiso de los estudiantes para practicar por fuera de las sesiones. Se requiere de una participación constante para lograr los resultados esperados.

## **3. Resultados de aprendizaje**

El curso tiene como objetivo:

- 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

| Clase                                        | Fecha      | Tema                                          |
|----------------------------------------------|------------|-----------------------------------------------|
| Clase 1                                      | Enero 20   | Introducción a Python                         |
| Clase 2                                      | Enero 27   | Introducción a Python II                      |
| Clase 3                                      | Febrero 3  | Exploración de objetos iterables              |
| Clase 4                                      | Febrero 10 | Condicionales y Loops: controles              |
| Clase 5                                      | Febrero 17 | Más Loops y Funciones I                       |
| Clase 6                                      | Febrero 24 | TALLER 1                                      |
| Clase 7                                      | Marzo 3    | Continuación Taller 1                         |
| Clase 8                                      | Marzo 10   | Bases de datos I                              |
| <b>Semana de receso: marzo 16 a marzo 21</b> |            |                                               |
| Clase 9                                      | Marzo 24   | Bases de datos II                             |
| <b>Semana Santa: marzo 30 a abril 4</b>      |            |                                               |
| Clase 10                                     | Abril 7    | Bases de datos III                            |
| Clase 11                                     | Abril 14   | Visualización de datos                        |
| Clase 12                                     | Abril 21   | Repaso de Pandas y Matplotlib                 |
| Clase 13                                     | Abril 28   | TALLER 2                                      |
| Clase 14                                     | Mayo 5     | Econometría e Introducción a Machine Learning |
| Clase 15                                     | Mayo 12    | Introducción a Machine Learning II            |
| Clase 16                                     | Mayo 19    | TALLER 3                                      |

## **Semana 1**

### **Clase 1: Introducción a Python**

- ¿Qué es? ¿Para qué?
- Ayuda con instalación en computadores personales.
- Funciones internas de Python.
- Operaciones matemáticas.

## **Semana 2**

### **Clase 2: Introducción a Python II**

- Estructuras de datos básicas: int, str, float, listas.
- Asignar objetos, mutarlos y borrarlos.
- Métodos y atributos.
- Indexación de listas

## **Semana 3**

### **Clase 3: Exploración de objetos iterables**

- Más objetos iterables: diccionarios y sus métodos.
- Funciones de identificación y conversión.

## **Semana 4**

### **Clase 4: Condicionales y Loops: controles**

- Condicionales.
- If, elif y else.
- For y While
- Controles.

## **Semana 5**

### **Clase 5: Más Loops y Funciones I**

- Listas Comprensivas.
- Definición de funciones
- Parámetros de funciones
- Documentación

## **Semana 6**

### **Clase 6: TALLER 1**

- Instalación e importación de paquetes.
- Numpy.

## **Semana 7**

### **Clase 7: Continuación Taller 1**

- Zip
- Repaso

## **Semana 8**

### **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**

### **Semana de Receso**

### **Semana 10 - octubre 11 última fecha para informar 30%**

### **Clase 9: Bases de datos II**

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

## **Semana 11**

### **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**

### **Clase 11: Visualización de datos**

- Matplotlib.
- Seaborn.

## **Semana 13**

### **Clase 12: Repaso de Pandas y Matplotlib**

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

## **Semana 14**

### **Clase 13: TALLER 2**

## **Semana 15**

### **Clase 14: Econometría e Introducción a Machine Learning**

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

## **Semana 16**

### **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**

### **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. En este curso, cada clase se dividirá en tres partes: en la primera parte el profesor introducirá técnicas, comandos y conceptos relacionados con el uso de Python y usará ejemplos ilustrados; en la segunda parte los estudiantes trabajarán individualmente (con apoyo de sus compañeros) en diversos ejercicios que contengan los conceptos explicados en la clase, y la tercera parte será la entrega individual de un ejercicio.

En el semestre se realizarán tres (3) talleres en clase sobre los temas que se abordan. Los estudiantes deben desarrollar estos talleres durante la clase, pero tendrán hasta la medianoche del domingo para entregarlos (la fecha de entrega puede estar sujeta a modificaciones). Para fomentar la integración de los estudiantes en el aula, la presentación de los talleres deberá realizarse obligatoriamente en 2 estudiantes, elegidos aleatoriamente para cada asignación de taller. La comunicación con su equipo es responsabilidad de cada estudiante, ya que, salvo causa mayor (retiro de alguno de los estudiantes del grupo, entre otras) la presentación de estos talleres de otra forma a la dispuesta generará una penalización de 1.0 unidades sobre la presentación de estos.

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 realizar una preparación previa del contenido de la clase según el programa descrito.

## 7. Evaluaciones

Los talleres se evaluarán teniendo en cuenta:

- a) Organización del código (estructura y comentarios),
- b) Que el código genere un resultado correcto o la lógica de la solución sea coherente y,
- c) Número de errores en el código.

Los talleres deben presentarse a través de Bloque Neón.

La valoración de las actividades del curso corresponde a:

| Evaluación               | Valoración (%) |
|--------------------------|----------------|
| Taller 1                 | 25             |
| Taller 2                 | 25             |
| Taller 3                 | 30             |
| Participación/Ejercicios | 20             |

- ✓ 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.

## 9. Fraude académico

Las conductas que se consideran fraude académico se encuentran en el artículo 4 del [Régimen Disciplinario](#).

La programación no es unívoca. La resolución de ejercicios es posible realizarse de diferentes maneras. Se espera que los estudiantes realicen los talleres y examen final de forma individual. Códigos que sean evidentemente iguales, serán penalizados al momento de calificación.

En caso de comprobarse fraude en las pruebas individuales y el examen final, los estudiantes serán penalizados con una nota de 0.0 independiente de la nota obtenida. Por fraude se considera compartir códigos o respuestas tanto de las pruebas como del examen final.

## 10. 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.