

¡Bienvenidos al curso Programación!

¡Bienvenidos al apasionante mundo de la programación en Python! Python es conocido por su sintaxis clara y legible, lo que lo convierte en la elección perfecta para principiantes y expertos por igual. A lo largo de este curso, exploraremos desde los conceptos fundamentales hasta temas más avanzados, permitiéndote adquirir las habilidades necesarias para desarrollar aplicaciones, automatizar tareas y resolver problemas de manera eficiente.

Nuestro objetivo es crear un ambiente de aprendizaje colaborativo y enriquecedor. No importa si estás dando tus primeros pasos en la programación o ya tienes experiencia. ¡Prepárate para sumergirte en el fascinante universo de Python y descubrir las infinitas posibilidades que te ofrece!

1. Información de la asignatura

Nombre de la asignatura: Programación

Código de la asignatura: PRE2013A45

Créditos académicos: 2

Semestre: 2024–I

Horario de clases: Grupo 02: Lunes 07:00 – 10:00 (3.6B) / Grupo 03: Lunes 13:00 – 16:00 (3.4B)

Profesor: Andrés Quintero Zea

Correo electrónico: andres.quintero27@eia.edu.co

Horario de atención: [Horario de Oficina]

2. Descripción de la asignatura

Introducción al Curso:

En este curso de programación en Python, exploraremos los fundamentos del lenguaje y su aplicación en la ciencia computacional y la ciencia de datos. A lo largo de nuestras lecciones, no solo adquirirás habilidades esenciales de programación, sino que también descubrirás cómo Python se ha convertido en una herramienta clave en la resolución de problemas y el análisis de datos. Este enfoque introductorio te proporcionará una visión integral, permitiéndote no solo aprender a programar, sino también comprender cómo aplicar estas habilidades en contextos prácticos y desafiantes.

Contenido del Curso:

El contenido de este curso está diseñado para proporcionar una experiencia integral de aprendizaje. Iniciaremos con los conceptos básicos, como variables, estructuras de control y funciones, avanzando progresivamente hacia temas más complejos como la manipulación de datos y programación orientada a objetos. Además, exploraremos bibliotecas populares y prácticas recomendadas que te permitirán abordar proyectos del mundo real con confianza. Para esto, el curso lo desarrollaremos en cuatro módulos:

1. Sintaxis y librerías básicas
2. Estructuras de datos y librerías adicionales
3. Programación Orientada a Objetos
4. Aplicaciones de Ciencia computacional y Ciencia de datos

Metodología de Enseñanza:

El curso se desarrollará mediante un enfoque práctico y participativo. Cada sesión combinará explicaciones teóricas con ejercicios prácticos. Tu participación activa mediante preguntas y discusiones en clase serán muy importantes para garantizar un aprendizaje efectivo.

Evaluación:

Cada módulo del curso involucra un evento evaluativo, que consiste en el desarrollo de un Notebook que brinde solución a los problemas planteados al inicio de cada módulo.

| EVALUACIÓN | PONDERACIÓN | FECHA |
|------------|-------------|------------|
| Trabajo 1 | 20 % | Febrero 24 |
| Trabajo 2 | 25 % | Marzo 23 |
| Trabajo 3 | 25 % | Abril 27 |
| Trabajo 4 | 30 % | Junio 1 |

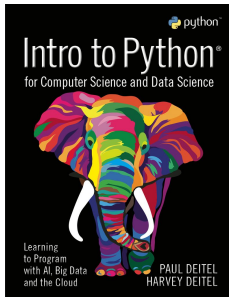
Recursos de apoyo:

El material de clase lo podrás encontrar en un repositorio en [GitHub](#) que he creado para tal fin. Los recursos los iré actualizando a medida que avancemos en el desarrollo del semestre, por lo que es importante que revises la vigencia de los mismos en la columna `Last commit message`, en el cual aparecerá `Updated 2024-1.`

3. Bibliografía

La bibliografía principal para este curso será *“Intro to Python for Computer Science and Data Science: Learning to Program with AI, Big Data and the Cloud”* escrito por **Harvey Deitel y Paul J. Deitel**. Este libro ofrece una

sólida introducción al mundo de Python, abarcando desde los conceptos fundamentales hasta aplicaciones avanzadas en inteligencia artificial, big data y la nube.



Norman S. Nise, *Control Systems Engineering*, 8th Edition, Wiley, 2019.

4. Políticas de entregas

Las entregas de los trabajos se rigen por las normas que se listan a continuación, procura siempre tenerlas en cuenta para evitar malentendidos.

- Se puede trabajar en parejas, pero cada uno de los módulos debe presentarse **con un compañero diferente**.
- Aplican hasta **dos** días de gracia por cada trabajo.
- La entrega se recibe únicamente a través del **Campus Digital** (Moodle).
- La fecha y hora de entrega serán las reportadas por el servidor, por lo que se sugiere realizar el envío con suficiente antelación a la hora de corte.
- La hora límite para la presentación a tiempo es la medianoche (23:59) en la fecha de entrega.
- Cada entrega debe ser presentada en un **único archivo** ZIP (o 7z) en el que se deben incluir el archivo PDF con el informe escrito (si está hecho *a mano*, debe ser escaneado con **excelente** calidad que permita una lectura fluida, en caso contrario se asignará **nota de 0.0**, ya que **si no lo puedo leer, no lo puedo calificar**) y los archivos de MATLAB que fueron usados.
- El archivo se debe nombrar de la siguiente manera: **PRG NombreApellido1 NombreApellido2 Trabajo No. X**, en caso de no hacerlo, **se descontará 0.2 de la nota obtenida**.

Entregas tardías*:

- El plazo límite para la entrega puntual es a las 23:59 en la fecha de vencimiento. Los días de retraso se cuentan en períodos completos de 24 horas. La entrega realizada entre las 00:00 y las 23:59 del día siguiente se considera entregada con **un día de retraso** y así sucesivamente. Por lo que se sugiere al estudiante no dejar la entrega para última hora.
- A la nota del trabajo entregado tarde se le aplicará una penalización de **0.5 por cada día de retraso**.

*Esta política de entregas tardías es en gran medida una copia de la original de Julie Zelenski para el curso **CS107** en la Universidad de Stanford.

- Al inicio del semestre, cada estudiante con una *bolsa* con tres **días de gracia** (ver **explicación** detallada más adelante) que puede utilizar para evitar la penalización por hacer la entrega de los informes de forma tardía.
- Siempre se aplicará el **principio de favorabilidad**, es decir, no tienes que indicar en la entrega si haces uso de días de gracia. Siempre aplicaré el máximo de días de gracia antes de penalizar.
- Para trabajo en parejas, ambos deben contar con días de gracia disponibles, ya que estos son personales e intransferibles. En caso de hacer uso de los mismos, se descontará a cada uno la misma cantidad.
- Cada tarea tiene una **fecha límite estricta**, generalmente dos o tres días después de la fecha de vencimiento original. No se aceptarán entregas tardías (con o sin penalización) después de la fecha límite estricta.
- Las prórrogas otorgadas por mí solo se considerarán después de haber utilizado todos los días de gracia y solo se otorgarán en **situaciones excepcionales**.

Días de gracia

Los días de gracia son “**prórrogas autoconcedidas**”. Como profesor, reconozco que a veces te enfrentas a cursos desafiantes y actividades extracurriculares que pueden ser un panorama agobiante en el que incluso los planes más cuidadosos pueden descarrilarse por un evento imprevisto.

Cuando sucede lo inesperado: te enfermas de gripe, se cae tu conexión a internet, se te “va la luz”, eliminas un archivo crítico (ojalá no sea win32), una situación personal se agrava; normalmente tendrías que solicitarme una prórroga. En lugar de eso y apelando a la equidad, tú y todos tus compañeros tienen el privilegio de otorgarse una extensión sin que medie ningún tipo de negociación. Un día de gracia amplía tu plazo de entrega en 24 horas.

Los días de gracia son tu medio para cubrir problemas reales: enfermedades, portátiles robados, accidentes de bicicleta, etc. No pretendo que los uses para cubrir viajes, conciertos, “guayabos” o una mala planificación, pero puedes usarlos como quieras y no ejerceré ningún control sobre su uso. Puedes usar todos tus días de gracia para una entrega, distribuirlos entre varias o, mejor aún, acumularlos para cuando se te presente una verdadera situación excepcional.

Aquí te pongo dos ejemplos de su funcionamiento:

- Un estudiante, llamémosle Ernesto[†], tiene sus **cuatro** días de gracia disponibles y debe hacer entrega del informe un martes a las 23:59. Son las 23:50, ingresa a Moodle de manera normal, pero al momento de darle click al botón de entrega, *Roberto Manchitas* (su perro) muerde el cable de energía del router que le provee de internet, por lo que Ernesto debe encender el punto de acceso de su teléfono móvil y compartir la conexión de datos con el computador. La conexión a internet se restablece, pero ya son las 00:00 del miércoles, por lo que el servidor de Moodle registra la entrega como retrasada. El informe de Ernesto es excepcional y, sin necesidad de que Ernesto me avise de sus problemas con Roberto Manchitas, yo procedo a calificar su informe y le asigno una nota de 5.0. Producto de esto, Ernesto tiene ahora

[†]La importancia de llamarse Ernesto, libro de Óscar Wilde.

en su bolsa **tres** días de gracia.

- Para la misma entrega, Cecilia[†] también hizo un informe excepcional, pero solo tiene disponible **un** día de gracia en su bolsa. El día de la entrega (martes), Cecilia salió a cenar con Gwendolen[§], su mejor amiga, confiando en llegar temprano a su casa para poder hacer la entrega del informe a tiempo. Pero cuando salían del restaurante, se encuentran con Jack y Algy[‡], quienes las invitan a un *plancito tranqui* que desemboca en una *pool party* en San Jerónimo. Cecilia regresa a su casa el miércoles a las 23:58 y procede a hacer la entrega del informe, que en el servidor queda registrada a las 00:03 del jueves. Yo procedo a calificar su informe y le asigno una nota de 4.5, ya que desconté de su bolsa el último día de gracia disponible (que cubrió el miércoles) y penalicé la nota con 0.5, ya que la entrega efectiva se hizo el jueves, según el reporte del servidor.

5. Calendario

Los números entre paréntesis se refieren a las secciones del libro de Deitel & Deitel.

| SESIÓN | FECHA | TEMA A ABORDAR |
|--------|--------|---|
| 1 | 22-ene | Python para CS/DS – Editor/IDE de Python (1.7, 1.8, 1.10, 1.14) |
| 2 | 29-ene | Programación en Python (Ch. 2) |
| 3 | 5-feb | Estructuras de control y desarrollo de programas (Ch. 3) |
| 4 | 12-feb | Funciones (Ch. 4) |
| 5 | 19-feb | Listas y tuplas (Ch. 5) |
| 6 | 26-feb | Diccionarios y conjuntos (Ch. 6) |
| 7 | 4-mar | Programación orientada a arreglos (Ch. 7) |
| 8 | 11-mar | Manejo de cadenas de texto (Ch. 8) |
| 9 | 18-mar | Archivos y excepciones (Ch. 9) |
| 10 | 1-abr | Programación orientada a Objetos I (10.1 – 10.6) |
| 11 | 8-abr | Programación orientada a Objetos II (10.7 – 10.11) |

[†]Otro personaje del mismo libro.

[§]Un personaje más.

[‡]Los últimos dos personajes, lo prometo.

| SESIÓN | FECHA | TEMA A ABORDAR |
|--------|--------|--|
| 12 | 15-abr | Programación orientada a Objetos III (10.12 – 10.16) |
| 13 | 22-abr | Recursividad I (11.1 - 11.9) |
| 14 | 29-abr | Recursividad II (11.10 - 11.15) |
| 15 | 6-may | Clasificación (15.1 - 15.3) |
| 16 | 20-may | Regresión y agrupamiento (15.4 - 15.7) |