

## Contenido

Actividades Iniciales.....	2
<b>Objetivo general:</b> .....	2
<b>Objetivos específicos:</b> .....	2
1. Test Estilos de aprendizaje .....	2
2. Encuesta .....	2
3. Equipos de trabajo .....	3
4. Introducción a Google Colab.....	3
Usos en la resolución de problemas de Análisis Numérico .....	3
Ventajas.....	4
5. Librerías de Python en el Cálculo Numérico y Simbólico .....	4
Cálculo Numérico: principales librerías .....	4
Cálculo Simbólico: principales librerías .....	4
Usos en el Análisis Numérico .....	5
Fuentes para ampliar las definiciones .....	5
6. Actividades .....	5
<b>Actividad: Individual</b> .....	5
A) Primeros pasos con Google Colab.....	5
<b>Actividad: Grupal</b> .....	6
B) Librerías de Python en el Cálculo Numérico y Simbólico .....	6
Actividad 1 Librerías clave para el Análisis Numérico con Python .....	6
Actividad 2 Cálculo de Derivadas, límites e integrales.....	6
Actividad 3 Introducción a la librería Matplotlib .....	7
Exploración práctica .....	7
<b>Formato de entrega</b> .....	7
<b>Criterios de acreditación</b> .....	8

## Actividades Iniciales

### Objetivo general:

- Iniciar a los estudiantes en el uso de herramientas digitales colaborativas aplicadas a la programación en Python para el abordaje de contenidos fundamentales del Análisis Numérico, promoviendo el trabajo en equipo, la reflexión sobre los propios estilos de aprendizaje y el desarrollo de competencias digitales.

### Objetivos específicos:

- Identificar los estilos de aprendizaje personales como punto de partida para mejorar la experiencia formativa.
- Releva el conocimiento previo en relación con herramientas de cálculo, programación e inteligencia artificial.
- Conformar equipos de trabajo colaborativo para el desarrollo de prácticas durante el cuatrimestre.
- Explorar y familiarizarse con Google Colab como entorno de desarrollo colaborativo para Python.
- Investigar el uso y la aplicabilidad de las principales librerías de Python en el cálculo numérico y simbólico.
- Utilizar cuadernos Jupyter como medio para la documentación y entrega de actividades académicas con estructura clara y bien argumentada.
- Promover la visualización como recurso para la interpretación de resultados en problemas numéricos.

### 1. Test Estilos de aprendizaje

Te invitamos a que realices el test con el propósito de identificar los estilos de aprendizaje variados con los que trabajarán los profesores durante el proceso formativo en cada una de las carreras. El estilo de aprendizaje hace referencia a las distintas maneras en que una persona puede aprender; describen las condiciones en las que un individuo encuentra en la mejor situación para aprender, o qué estructura necesita para mejorar el proceso de aprendizaje. Es importante ya que permite identificar la manera distintiva en que las personas resuelven problemas, es decir, la respuesta a estímulos e información. También, es posible analizar los estilos de aprendizaje a partir del comportamiento característico de una persona, desde lo afectivo, fisiológico y cognitivo, lo cual permite identificar la manera en que el alumno interactúa con un entorno de aprendizaje.

Responde el test en el siguiente enlace:

<https://docs.google.com/forms/d/e/1FAIpQLSf5UOHY9oKKaJSszg9QqJgoqYXUWtAfvd-7IPc8lkcFntPCg/viewform?usp=sharing&oid=103876299178824533060>

### 2. Encuesta

Realiza la siguiente encuesta sobre herramientas de cálculo, programación e Inteligencia artificial.

<https://docs.google.com/forms/d/e/1FAIpQLSdZlqgYs5Z95dianvxwzPfjJopvXRNAyeefTNXxZLUFmIKnoA/viewform?usp=sharing&oid=103876299178824533060>

### 3. Equipos de trabajo

Durante el cursado trabajaremos en equipos de como máximo 5 estudiantes (sin excepción). Conformaran los grupos y completaran el siguiente formulario debiendo respetar la fecha límite para realizarlo.

Los correos electrónicos deben ser @gmail.com ya que trabajaremos con Google Drive y Google Colab. Cada grupo deberá contar con referente del cual indicaran también los datos.

[https://docs.google.com/forms/d/e/1FAIpQLSeDslcMd-XsD\\_5cgct-nBcuKRouscivqJl4hODb0wnqHDeWA/viewform?usp=sharing&oid=103876299178824533060](https://docs.google.com/forms/d/e/1FAIpQLSeDslcMd-XsD_5cgct-nBcuKRouscivqJl4hODb0wnqHDeWA/viewform?usp=sharing&oid=103876299178824533060)

### 4. Introducción a Google Colab

Google Colaboratory (Colab) es una herramienta gratuita basada en la nube que permite escribir y ejecutar código en lenguaje Python directamente desde un navegador web. Desarrollada por Google, está integrada con Google Drive, lo que facilita el almacenamiento automático de los documentos (notebooks) y la colaboración en tiempo real con otros usuarios.

#### Funciones principales

- ⊕ Ejecución de código Python: Se pueden crear celdas de código para realizar cálculos, visualizaciones y simulaciones.
- ⊕ Uso de texto enriquecido con Markdown: Permite documentar el trabajo de forma clara con títulos, listas, fórmulas matemáticas (LaTeX), enlaces, imágenes, entre otros.
- ⊕ Almacenamiento en la nube: Guarda automáticamente los cuadernos en Google Drive.
- ⊕ Acceso a recursos avanzados: Ofrece la posibilidad de utilizar GPU y TPU sin costo adicional para cálculos de alto rendimiento.
- ⊕ Colaboración en tiempo real: Varias personas pueden trabajar simultáneamente en un mismo cuaderno, al igual que en Google Docs.
- ⊕ Importación de bibliotecas: Se pueden instalar y utilizar bibliotecas de Python como NumPy, SymPy, Matplotlib, SciPy y pandas, entre muchas otras.

### Usos en la resolución de problemas de Análisis Numérico

En el contexto de materias universitarias como Análisis Numérico, Google Colab se convierte en un entorno muy poderoso para:

- ⊕ Explorar funciones matemáticas mediante gráficos y análisis de propiedades.
- ⊕ Resolver ecuaciones diferenciales o integrales de forma simbólica o numérica.
- ⊕ Simular y visualizar conceptos teóricos como límites, derivadas, integrales o series.
- ⊕ Implementar algoritmos numéricos para la resolución de ecuaciones, interpolación, integración numérica, etc.
- ⊕ Analizar errores y convergencia en métodos de aproximación.

- ⊕ Crear informes interactivos, combinando cálculos, gráficos y explicaciones teóricas en un solo documento.

## Ventajas

- ⊕ Promueve el aprendizaje activo mediante la experimentación directa con los algoritmos.
- ⊕ Facilita el desarrollo de competencias digitales en entornos colaborativos.
- ⊕ Mejora la comprensión de los contenidos matemáticos abstractos a través de la visualización.
- ⊕ Permite la entrega, corrección y seguimiento de actividades de forma eficiente en plataformas como Moodle.

## 5. Librerías de Python en el Cálculo Numérico y Simbólico

Python es un lenguaje ampliamente adoptado en la enseñanza y aplicación del **análisis numérico** y el **cálculo simbólico**, gracias a un ecosistema de librerías especializadas que permiten implementar algoritmos, resolver ecuaciones, simular modelos y visualizar resultados de forma sencilla y potente.

### Cálculo Numérico: principales librerías

#### 1. NumPy (Numerical Python)

- ⊕ **Uso principal:** operaciones con vectores, matrices y álgebra lineal básica.
- ⊕ **Aplicaciones:** resolución de sistemas de ecuaciones lineales, aproximaciones numéricas, métodos iterativos.
- ⊕ **Ventaja:** alto rendimiento en operaciones con arrays multidimensionales.

#### 2. SciPy (Scientific Python)

- ⊕ **Uso principal:** biblioteca científica de nivel superior que se apoya en NumPy.
- ⊕ **Aplicaciones:** integración numérica, optimización, interpolación, derivación, transformadas, entre otros.
- ⊕ **Módulos útiles:** `scipy.integrate`, `scipy.optimize`, `scipy.interpolate`.

#### 3. Matplotlib

- ⊕ **Uso principal:** visualización de datos y funciones matemáticas.
- ⊕ **Aplicaciones:** representación gráfica de funciones, análisis visual de soluciones numéricas.
- ⊕ **Complemento de aprendizaje:** permite a los estudiantes interpretar gráficamente los resultados de algoritmos.

#### 4. Pandas

- ⊕ **Uso principal:** manipulación y análisis de datos tabulares.
- ⊕ **Aplicaciones:** tratamiento de resultados numéricos, análisis de errores, gestión de conjuntos de datos experimentales.

### Cálculo Simbólico: principales librerías

#### 1. SymPy (Symbolic Python)

- ⊕ **Uso principal:** cálculo simbólico exacto.

- ⊕ **Aplicaciones:** derivación e integración simbólica, resolución de ecuaciones, álgebra, simplificación de expresiones.

## 2. SageMath (basado en Python)

- ⊕ **Uso principal:** plataforma integrada para matemática avanzada.
- ⊕ **Aplicaciones:** combina herramientas numéricas y simbólicas para álgebra lineal, geometría, teoría de números, ecuaciones diferenciales, etc.
- ⊕ **Nota:** aunque no es una librería en sí, utiliza Python como lenguaje base y puede integrarse con Jupyter Notebooks.

## Usos en el Análisis Numérico

Estas herramientas permiten implementar con eficacia temas clave de análisis numérico como:

- ⊕ Métodos de bisección, Newton-Raphson y punto fijo.
- ⊕ Interpolación polinómica y spline.
- ⊕ Integración numérica (regla del trapecio, Simpson).
- ⊕ Derivación numérica.
- ⊕ Resolución numérica de ecuaciones diferenciales ordinarias.
- ⊕ Análisis de estabilidad y error de métodos aproximados.

## Fuentes para ampliar las definiciones

### 1. Documentación oficial:

- NumPy: <https://numpy.org/doc/>
- SciPy: <https://docs.scipy.org/doc/scipy/>
- SymPy: <https://docs.sympy.org/>
- Matplotlib: <https://matplotlib.org/stable/contents.html>
- Pandas: <https://pandas.pydata.org/docs/>

## 6. Actividades

### Actividad Individual

#### A) Primeros pasos con Google Colab

1. Ingresá al siguiente enlace y leé atentamente el contenido sobre Google Colab:  
👉 [Guía introductoria a Google Colab - Dialéktico](#)
2. Luego, mirá el siguiente tutorial en video para reforzar los conceptos:  
📺 [Video tutorial: Primeros pasos con Google Colab](#)

3. Una vez finalizadas ambas instancias, ingresá al aula virtual y completá el cuestionario titulado "Primeros pasos con Google Colab".

Esta actividad te permitirá familiarizarte con la herramienta que utilizaremos a lo largo del curso para programar en Python de forma colaborativa y en la nube.

## Actividad Grupal

### B) Librerías de Python en el Cálculo Numérico y Simbólico

Python cuenta con múltiples librerías que facilitan la implementación de algoritmos numéricos y simbólicos. En esta actividad vas a investigar y experimentar con cuatro de las más utilizadas en la materia: NumPy, SymPy, SciPy y Matplotlib.

#### Actividad 1 Librerías clave para el Análisis Numérico con Python

Para cada una de las siguientes librerías (NumPy, SymPy, SciPy, Matplotlib), respondé en una celda de texto las siguientes preguntas. Podés consultar documentación oficial, tutoriales, videos o utilizar una inteligencia artificial como asistente de búsqueda:

1. ¿Cuál es el propósito principal de esta librería?
2. ¿Qué tipo de objetos o estructuras de datos introduce (por ejemplo: arrays, matrices, expresiones simbólicas)?
3. Menciona al menos dos funciones o métodos que consideres útiles para Análisis Numérico.
4. Especificá una situación práctica o ejemplo de problema donde esta librería sería especialmente útil.
5. Agregá la fuente que consultaste (link o referencia).

*Tip:* Podés organizar la información en un cuadro comparativo o usar títulos para cada librería.

#### Actividad 2 Cálculo de Derivadas, límites e integrales

1. Investiga cómo se calculan derivadas simbólicas utilizando SymPy. Utiliza la función `diff` para calcular derivadas simples y parciales. Proporciona un ejemplo para cada caso.
2. Explora cómo se calcula un límite simbólico usando SymPy. Investiga la función `limit` y proporciona un ejemplo de cómo se calcularía el límite de una función cuando una variable tiende a un valor específico.
3. ¿Qué ventajas ofrece el cálculo simbólico de límites con SymPy en comparación con hacerlo manualmente? Reflexiona sobre cómo SymPy maneja el cálculo simbólico en límites complicados.
4. Investiga algún caso en el que calcular el límite de una función simbólicamente sea más útil que evaluarlo numéricamente. Explica por qué el enfoque simbólico es necesario en ese caso.

- Investiga la función `integrate` de SymPy y explica cómo se utilizan para calcular integrales indefinidas y definidas. Proporciona un ejemplo de cada tipo de integral (indefinida y definida) resuelto simbólicamente.

### Actividad 3 Introducción a la librería Matplotlib

Matplotlib es una librería fundamental para la visualización de datos y resultados numéricos en Python. A través de esta actividad, vas a explorar su utilidad, aprender sus conceptos básicos y generar tus primeros gráficos.

Buscá información sobre Matplotlib utilizando fuentes confiables (pueden ser artículos, tutoriales, documentación oficial o una herramienta de inteligencia artificial como ChatGPT). Respondé las siguientes preguntas en una celda de texto:

- ¿Qué es Matplotlib y para qué se utiliza?
- ¿Cuál es la diferencia entre `matplotlib.pyplot` y otras formas de uso de la librería?
- ¿Qué tipos de gráficos se pueden generar con Matplotlib?
- ¿Qué ventajas ofrece esta librería para representar resultados numéricos en Ingeniería?
- Mencioná al menos dos funciones clave y explicá qué hacen.
- Compartí la fuente que utilizaste (link o nombre del recurso).

### Exploración práctica

Creá un gráfico simple que represente una función matemática, por ejemplo:

$$f(x) = x^2$$

$$f(x) = \sin(x)$$

- o un conjunto de puntos generados con NumPy

Personalizá el gráfico agregando:

- Título
- Etiquetas en los ejes
- Color y estilo de línea
- Leyenda (si corresponde)

Comentá el código para explicar qué hace cada parte.

### Pregunta de reflexión

¿Por qué creés que la visualización es importante al trabajar con métodos numéricos? ¿Qué aporta un gráfico que no aporta una tabla de valores?

### Formato de entrega

Para la resolución de este trabajo práctico se requiere la utilización de cuadernos de Jupyter. A continuación, se detallan las instrucciones para la correcta presentación de las actividades:

#### Formato de Presentación:

- Crear un cuaderno de Jupyter titulado `GrupoX_TP_0`, donde X corresponde al número de tu grupo.

- En este cuaderno, se deberá desarrollar las consignas con los ejemplos solicitados. Puedes apoyarte en herramientas de Inteligencia Artificial si lo consideras necesario. Cada cuaderno deberá incluir los nombres de los integrantes del grupo al inicio y estar bien organizado con títulos y explicaciones claras.
- Compartir el cuaderno con el docente a la dirección [claudiagarcia@gfe.frre.utn.edu.ar](mailto:claudiagarcia@gfe.frre.utn.edu.ar) y enviar el link de acceso en la tarea “**Librerías de Python en el Cálculo Numérico y Simbólico**”.

## Criterios de acreditación

Para que la actividad sea considerada **aprobada**, se tendrán en cuenta los siguientes criterios:

1. **Participación completa en las actividades iniciales:**
  - Realización del test de estilos de aprendizaje.
  - Respuesta a la encuesta diagnóstica.
  - Registro en el formulario de equipos de trabajo.
2. **Uso adecuado de las herramientas propuestas:**
  - Creación y uso de Google Colab para el desarrollo de las consignas.
  - Presentación del cuaderno Jupyter bien estructurado, con uso de Markdown para explicaciones claras.
3. **Calidad en el desarrollo de las actividades:**
  - Respuestas claras, correctas y fundamentadas a las consignas.
  - Inclusión de ejemplos funcionales y comentados en código Python.
  - Reflexión crítica en las preguntas propuestas.
4. **Organización y presentación:**
  - Entrega en tiempo y forma según lo indicado.
  - Identificación correcta del grupo y sus integrantes.
  - Compartir correctamente el cuaderno con el docente y por el canal correspondiente.
5. **Trabajo colaborativo:**
  - Participación activa de todos los miembros del grupo en las actividades grupales.