

# 1er Parcial de Introducción a la Programación en Python

## Matemática para Economistas III

### IDEI-UNGS.

16 de mayo de 2022

*Tenés que combatir el algoritmo,  
porque vos buscaste dos o tres  
cosas y te tiran por la cabeza  
70.000 idioteces que no te  
interesan*

---

Adrián Dárgelos

Docente: Mateo Suster — msuster@campus.ungs.edu.ar —  
Fecha límite de entrega: Lunes 9 de Mayo 23:59 hs

#### Pautas del examen

El examen deberá ser realizado a partir de los siguientes lineamientos:

1. Se debe resolver individualmente.
2. Se deberá entregar el código de Python utilizado en formato **.ipynb exclusivamente**.
3. **Nombrar** el archivo con el apellido y nombre del estudiante (IMPORTANTE).
4. Se evaluará positivamente a la inclusión de las buenas prácticas de programación vistas en clase, así como la **prolijidad** y **originalidad** de las soluciones propuestas.
5. Es válido (y preferible) solicitar ayuda entre pares (alumnos y/o docente) por Slack antes que por otro medio. Su consulta no molesta (al contrario, enriquece).
6. Utilizar código ajeno no está prohibido, siempre y cuando se explicita la fuente correspondiente. Se podrá agregar todas las aclaraciones extras que se deseen o se consideren relevantes.
7. La entrega deberá realizarse a través de mail a la casilla msuster@campus.ungs.edu.ar con el asunto "MPE III - Parcial de Python".
8. Último, pero no menos importante: antes de entregar, revise que el código se ejecute de principio a fin sin errores !

#### Ejercicios

1. Elegir una ecuación diferencial (de la práctica o planteada por usted) y presentarla en lenguaje Mark-down.
2. Justificar la elección realizada en el punto anterior (explicación mínima, corta y simple de porqué le interesa resolver la ecuación elegida).

3. Resolver la ecuación diferencial con la librería SymPy. Discutir la estabilidad dinámica de la solución.
4. Resolver nuevamente la misma ecuación diferencial modificando las condiciones iniciales. Visualizar en **un mismo gráfico** la evolución a lo largo del tiempo de la ecuación elegida con las dos condiciones iniciales propuestas. Explicar porqué y cómo se modifica el comportamiento de la función. Tener en cuenta que quizás sea necesario ir variando el rango de tiempo en donde se evalúa la ecuación para lograr una mejor visualización. Bonus: graficar también el equilibrio intertemporal.
5. Definir tres (3) listas distintas con al menos 5 elementos de (inventar lo menos posible):
  - Nombres de países
  - Población de países
  - Casos confirmados acumulados de COVID-19 al día de la fecha

A partir de dichos objetos, escribir un programa que evalúe si ‘Argentina’ se encuentra dentro de la lista de países. Luego, en caso de ser cierto, el programa debe incluir un bucle while que estime la tasa de casos confirmados cada millón de habitantes (u otra métrica de interés) para todos los países y guarde dichos valores en una nueva lista. En caso contrario, el programa deberá imprimir por pantalla alguna explicación de porqué no realizó los cálculos. Explicar el procedimiento y evaluar el funcionamiento del programa con una lista que incluya ‘Argentina’ y con otra que lo excluya.

6. Encapsular el programa del cálculo anterior en una función que devuelva la métrica elegida **si y solo si todas las listas de input** poseen el mismo largo. En caso contrario, la función debe devolver cuál es la diferencia máxima de longitud entre las listas. Evaluar la función con diferentes listas para revisar que corra sin fallas. Denominar la función con un nombre representativo del algoritmo. **Nota:** para calcular la discrepancia de longitud puede valerse de funciones de máximo y mínimo, como `max()` o `min()`.
7. Hacer un gráfico a elección para visualizar las tasas calculadas, incluyendo en el eje x el nombre de cada país. **Nota:** buscar en la documentación de la librería matplotlib un gráfico adecuado (por ejemplo, de barras) y citar la fuente.