

# Guía para el Proyecto de Interpolación con Splines Cúbicos

## Introducción

Esta guía tiene como objetivo facilitar la realización de un proyecto de interpolación utilizando splines cúbicos, con un enfoque en la creación y recreación de caricaturas. Esta iniciativa está dirigida a los estudiantes matriculados en la asignatura MAT156 - Análisis Numérico y, posteriormente, a los estudiantes de la asignatura INF243 - Métodos Numéricos (debido al cambio en la malla curricular) del programa de Informática de la Universidad Mayor de San Andrés.

## Beneficios del Proyecto

### Desarrollo de Habilidades Analíticas y Numéricas:

Al participar en este proyecto, los estudiantes tendrán la oportunidad de aplicar y mejorar sus habilidades en análisis numérico, lo que les proporcionará una base sólida para futuras investigaciones y aplicaciones prácticas.

### Aplicaciones en el Mundo Real:

La interpolación con splines cúbicos es una técnica ampliamente utilizada en diversas disciplinas, incluyendo la animación y la generación de gráficos. Los conocimientos adquiridos en este proyecto serán aplicables en situaciones del mundo real.

## Proceso del Proyecto

### Definición de Objetivos y Alcance

En esta fase inicial, nos concentramos en la claridad y precisión de nuestros objetivos. Imaginemos un plano cartesiano como un lienzo en blanco, donde nuestra meta es adquirir puntos de manera eficiente. Estos puntos serán meticulosamente organizados en dos listas, una para el eje X y otra para el eje Y. Este proceso nos permitirá, posteriormente, utilizar Python para crear una representación gráfica en un plano 2D. Antes de entrar en la transformación con los splines cúbicos, primero exploraremos cómo generar una gráfica básica utilizando estos datos iniciales.

### Transformación Mediante Splines Cúbicos

Avanzando hacia una etapa crítica, introducimos los potentes splines cúbicos. Esta fase representa una metamorfosis de nuestra representación inicial. Los splines cúbicos no solo transforman, sino que enriquecen. Permiten generar una multiplicidad de puntos adicionales de manera artificial, enriqueciendo nuestra representación. Estos puntos "extra" son como pinceladas adicionales en un lienzo, llenando detalles y matices que, de otra manera, podrían pasar desapercibidos. Esto se traduce en una mejora significativa de la definición de la gráfica, evidenciando la calidad de la primera representación en comparación con la enriquecida mediante splines cúbicos. Esta etapa es crucial, ya que establece la base para la próxima fase de interpolación.

### Composición Artística de la Caricatura

Con una comprensión clara de cómo se generan las gráficas y tras haber creado una abundancia de puntos a través de la interpolación, avanzamos hacia una fase emocionante: la composición artística de la caricatura. Es como si tuviéramos un repertorio de pinceladas, cada una representando una parte

única del conjunto total. Al combinar estas pinceladas, creamos una sinfonía visual, una amalgama de elementos que forman una representación coherente y vívida. Estas composiciones serán organizadas cuidadosamente en una secuencia, creando una narrativa visual que culminará en una caricatura final que refleje la esencia y la esencia de nuestro objetivo inicial.

### Conclusiones y Perspectivas Futuras

Al concluir este proyecto, nos encontramos no solo con una caricatura visualmente impresionante, sino también con un profundo entendimiento de los métodos de interpolación y la potencia de los splines cúbicos en el mundo de la representación gráfica. Esta experiencia no solo ha ampliado nuestro conocimiento práctico, sino que también ha fomentado habilidades esenciales en el campo del análisis numérico y la visualización de datos.

Este proyecto representa una oportunidad valiosa para los estudiantes de MAT156 y, posteriormente, de INF243, de explorar conceptos teóricos en un contexto práctico y emocionante. Además, sienta las bases para futuras investigaciones y aplicaciones en el campo de la visualización de datos y el análisis numérico.

### Pasos del proyecto

#### Paso 1: Importar librerías ->

Estos dos comandos importan las bibliotecas NumPy y Matplotlib.pyplot en tu programa, lo que te permitirá usar funciones y clases de estas bibliotecas más adelante en tu código.

```
---//code//---
```

```
# Importamos la biblioteca NumPy y la renombramos como np para abreviar.
```

```
import numpy as np
```

```
# Importamos la biblioteca Matplotlib y solo el módulo pyplot, que se renombra como plt para abreviar.
```

```
import matplotlib.pyplot as plt
```

```
---//fcode//---
```

#### Paso 2: Crear el arreglo con 'x' y 'y' y graficar ->

```
---//code//---
```

```
# Creamos un arreglo NumPy 'x' con los valores proporcionados.
```

```
x = np.array([6.45925419788322, 8.00000000000000, 8.49364223082215, 7.46868337453231,  
5.51194373979715, 4.05215385356616, 3.95897577572163, 5.01499399129298, 6.47478387752397])
```

```
# Creamos un arreglo NumPy 'y' con los valores proporcionados.
```

```
y = np.array([8.87559561819863, 8.00000000000000, 6.15790168106646, 4.55834467806866,  
4.38751820202035, 5.42800673795096, 7.55557284873443, 8.53394266610201, 8.86006593855787])
```

```
# Creamos una figura y un conjunto de ejes con una cierta dimensión.  
fig, axs = plt.subplots(figsize=(5, 5))  
  
# Graficamos una línea con los valores de 'x' y 'y' con un color hexadecimal (#050505).  
axs.plot(x, y, color='#050505')  
  
# Graficamos los puntos individuales de 'x' y 'y' con un color naranja (tab:orange).  
axs.scatter(x, y, color="tab:orange")  
  
# Mostramos el gráfico.  
plt.show()  
---//fcode//---
```

Paso 3: Final