

# Tarea Programada 3

---

Curso	Programación Avanzada en Web
Código	SC-701
Profesor	Luis Andrés Rojas Matey
Fecha y hora de entrega inicial	Martes 17 de junio a las 9:00 pm
Fecha y hora de entrega final	Martes 24 de junio a las 6:00 pm

## Introducción

Las funciones son elementos esenciales del álgebra. Con ellas, es posible representar relaciones entre elementos y encontrar valores cuando se utilizan en forma de ecuación.

## Objetivo

Aplicar los conocimientos adquiridos de utilizar un *minimal API* con la herramienta **ASP.NET Core Minimal API** del *framework* **.NET 8.0**.

## Especificaciones funcionales

Al ser una **Web API**, una vez se ejecuta, este *web service* podrá consumirse con cualquier cliente **HTTP/REST(ful)**, como lo son **Postman** o cualquier navegador web. Este servicio tendrá cuatro (4) *endpoints* que se deben poder acceder por medio del método/verbo **GET**:

- "/"
- "/lineal"
- "/cuad"
- "/exp"

### *Endpoint "/"*

Este *endpoint* hará un *redirect* al **UI** de **Swagger**, permitiendo observar la definición de los *endpoints* en una página web.

## Endpoint "/lineal"

Este permitirá identificar propiedades de una función lineal que define una recta de la forma  $f(x) = mx + b$ .

Para esto, permitirá los siguientes parámetros (del *query string*):

- **b** (obligatorio): valor numérico de la intersección con el eje **y**.
- **m** (opcional): valor numérico de la pendiente.
- **x1** (opcional): un valor numérico del eje **x** de la recta, asociado con **y1**.
- **y1** (opcional): un valor numérico del eje **y** de la recta, asociado con **x1**.
- **x2** (opcional): otro valor numérico del eje **x** de la recta, asociado con **y2**.
- **y2** (opcional): otro valor numérico del eje **y** de la recta, asociado con **x2**.
- **xml** (opcional): valor *boolean* indicando si el resultado se debe retornar en XML en vez de JSON. Por defecto (*default*) es **false**, es decir, por defecto se debe retornar un JSON.

A pesar que el único valor obligatorio es **b**, para que retorne un resultado correcto (**HTTP 200 Success**), el *request* debe incluir un valor en **m**, o bien, los dos puntos del plano cartesiano (**x1, y1**) y (**x2, y2**) representados por los cuatro valores de los ejes (**x1, y1, x2, y2**). En caso de que se provean los cinco valores, tomará prioridad **m**, obviando los demás cuatro. Si **m** no es proveído pero los otros cuatro sí, entonces se procederá a calcular el valor de **m** a partir de los dos puntos.

Una vez se tenga el valor del pendiente (**m**), el servicio devolverá un objeto (JSON) con cuatro campos (las tildes se pueden obviar y tanto los índices como los valores son de tipo *string*):

- **funcion**:  $f(x) = mx + b$ , con los respectivos valores de **m** y **b**.
- **pendiente**: **creciente** o **decreciente**.
- **interseccionConEjeX**: par ordenado (**x, 0**) con el valor correspondiente de **x**.
- **interseccionConEjeY**: par ordenado (**0, y**) con el valor correspondiente de **y**.

Por ejemplo, este *request*:

```
/lineal?b=3&m=-5
```

devolvería este *response* (**HTTP 200 Success**):

```
{
  "funcion": "f(x) = -5x + 3",
  "pendiente": "decreciente",
  "interseccionConEjeX": "(0.6, 0)",
  "interseccionConEjeY": "(0, 3)"
}
```

Así mismo, este otro *request*:

```
/lineal?b=3&m=-5&xml=true
```

retornaría este *response* (**HTTP 200 Success**):

```
<?xml version="1.0" encoding="utf-16"?>
<Lineal xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Funcion>f(x) = -5x + 3</Funcion>
  <Pendiente>decreciente</Pendiente>
  <InterseccionConEjeX>(0.6, 0)</InterseccionConEjeX>
  <InterseccionConEjeY>(0, 3)</InterseccionConEjeY>
</Lineal>
```

Además, este otro *request* que no contiene **m** ni los dos puntos necesarios para calcular la pendiente:

```
/lineal?b=3
```

debería devolver este *response* (**HTTP 400 Bad Request**):

```
{
  "error": "Debe proporcionar la pendiente (m) o dos puntos (x1, y1) y (x2, y2)."
}
```

## Endpoint "/cuad"

Este permitirá conocer las propiedades de una función cuadrática que define una parábola. Esta está definida como  $f(x) = ax^2 + bx + c$ , con los siguientes parámetros:

- **a** (obligatorio): valor numérico diferente de cero.
- **b** (obligatorio): valor numérico.
- **c** (obligatorio): valor numérico.
- **xml** (opcional): valor *boolean* indicando si el resultado se debe retornar en XML en vez de JSON. Por defecto (*default*) es *false*.

Si el valor de **a** es cero o alguno de los tres parámetros obligatorios (**a**, **b** y **c**) no son proveídos, se deberá retornar un **Bad Request**. Por el contrario, en el escenario válido, se deberá retornar un objeto JSON con los siguientes campos:

- **funcion:**  $f(x) = ax^2 + bx + c$ , con los valores correspondientes de **a**, **b** y **c**.
- **discriminante:**  $\Delta = \#$ , con **#** el valor calculado del discriminante.
- **ejeDeSimetria:**  $x = \#$ , con **#** como el valor de **x**.
- **concavidad:** **hacia arriba** o **hacia abajo**.
- **vertice:** par ordenado (**x**, **y**) con sus respectivos valores.
- **interseccionConEjeX:** **no hay**, (**x**, **y**) o (**x1**, **y1**) y (**x2**, **y2**).
- **interseccionConEjeY:** (**0**, **y**), con el valor respectivo de **y**.
- **ambito:** intervalo  $]-\infty, \#]$  o  $[\#, +\infty[$ , con el respectivo valor de **#**.
- **monotonias:** intervalos  $]\#, +\infty[$  y  $]-\infty, \#[$  donde **crece** y **decrece**, con los respectivos valores de **#**.

Ejemplos:

*Request:*

```
/cuad?a=0.5&b=-6&c=10
```

*Response:*

```
{
  "funcion": "f(x) = 0.5x² + -6x + 10",
  "discriminante": "Δ = 16",
  "ejeDeSimetria": "x = 6",
  "concavidad": "hacia arriba",
  "vertice": "(6, -8)",
  "interseccionConEjeX": "(10, 0) y (2, 0)",
  "interseccionConEjeY": "(0, 10)",
  "ambito": "[-8, +∞[",
  "monotonias": "crece en ]6, +∞[ y decrece en ]-∞, 6["
}
```

*Request:*

```
/cuad?a=0.5&b=-6&c=0&xml=true
```

Response:

```
<?xml version="1.0" encoding="utf-16"?>
<Cuadratica xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Funcion> $f(x) = 0.5x^2 + -6x + 10$ </Funcion>
  <Discriminante> $\Delta = 16$ </Discriminante>
  <EjeDeSimetria> $x = 6$ </EjeDeSimetria>
  <Concavidad>hacia arriba</Concavidad>
  <Vertice>(6, -8)</Vertice>
  <InterseccionConEjeX>(10, 0) y (2, 0)</InterseccionConEjeX>
  <InterseccionConEjeY>(0, 10)</InterseccionConEjeY>
  <Ambito>[-8, +∞[</Ambito>
  <Monotonias>crece en ]6, +∞[ y decrece en ]-∞, 6[</Monotonias>
</Cuadratica>
```

Request:

```
/cuad?a=0&b=-6&c=0
```

Response:

```
{
  "error": "Parámetro 'a' debe ser diferente de cero."
}
```

## Endpoint "/exp"

Al igual que los anteriores *endpoints*, este permite obtener algunas propiedades de una función exponencial de la forma  $f(x) = b^x$ , con los siguientes parámetros:

- **b** (obligatorio): valor numérico que indica la base, el cual debe ser mayor que cero (0) y diferente de uno (1).
- **xml** (opcional): valor *boolean* indicando si el resultado se debe retornar en XML en vez de JSON. Por defecto (*default*) es **false**.

El *response* exitoso incluiría estos datos:

- **funcion:**  $f(x) = b^x$ , con el valor correspondiente de **b**.
- **monotonia:** **creciente** o **decreciente**.

Ejemplos:

*Request:*

```
/exp?b=0.7
```

*Response*

```
{
  "funcion": "f(x) = 0.7^x",
  "monotonia": "decreciente"
}
```

*Request:*

```
/exp?b=1
```

*Response:*

```
{
  "error": "La base 'b' debe ser mayor que 0 y diferente de 1."
}
```

## Especificaciones técnicas

- El trabajo se debe realizar con el lenguaje de programación **C#**, la arquitectura **ASP.NET Core Minimal API** y el *framework* **.NET 8.0**.
- Debe contener un *solution* y un *project*, así como el *project* incluido en el *solution*.
- Los parámetros de los *requests* (los cuales son todos de "tipo" **GET**) deben ser enviados por medio del *query string*. Cualquier otro parámetro por otro medio (*header*, *form*, *body*, etc.) debe ser ignorado.
- Los *responses* pueden no contener palabras con tilde, es decir, las tildes se pueden obviar.

- En caso de que el *response* no sea **HTTP 200 Success** debido a una validación (por ejemplo, uno de los parámetros no cumple con la restricción del valor), entonces debe ser de tipo **HTTP 400 Bad Request** y retornar el mensaje de error en formato JSON.
- Se recomienda utilizar el editor [Visual Studio Code](#).

## Entregables

---

Al ser una tarea de carácter individual, se debe entregar un único archivo comprimido **ZIP** con el siguiente nombre: **TP3- [Carné].zip**. Ejemplo de nombre del archivo **ZIP**: **TP3-FH12345678.zip**.

El mismo debe contener lo siguiente:

- Todo el código fuente que incluya el archivo *solution* y la carpeta del *project*. Sin embargo, no debe contener los archivos compilados, es decir, excluir las carpetas **bin** y **obj**.
- **README.md**. La documentación en [Markdown](#) donde se indique lo siguiente:
  - Su nombre, carné y el enlace al repositorio si lo subió a [GitHub](#), [GitLab](#) o algún otro proveedor.
  - Los comandos de **dotnet** utilizados (**CLI**).
  - Páginas web donde halló posibles soluciones a problemas encontrados o *snippets* de código.
  - *Prompts* (consultas y respuestas) de los *chatbots* de IA ([Copilot](#), [Gemini](#), [ChatGPT](#), etc.) que haya utilizado.

## Evaluación

El proyecto será calificado según la rúbrica que se presenta en el programa del curso.