

Guía de Trabajo de Investigación

Unidad 2: Funciones

1. Contexto general

En el curso se consideran **5 notas**, cada una con un peso del 20 % en la nota final.

En particular, una de estas calificaciones corresponde al **EID de la Unidad 2: Funciones**, cuya evaluación se desglosa de la siguiente manera:

- **50 %:** Trabajo de investigación (Parte A y Parte B).
- **50 %:** Prueba grupal de resolución de problemas de funciones (al término de la unidad).

2. Resumen de ponderaciones Unidad 2

Actividad	Porcentaje en la Unidad 2	Porcentaje en la nota final del curso
Parte A: Investigación Matemática (video)	25 %	5 %
Parte B: Proyecto en Python (defensa presencial)	25 %	5 %
Prueba grupal de problemas de funciones	50 %	10 %
Total Nota Unidad 2	100 %	20 %

3. Organización del trabajo

La investigación que deben realizar durante la semana online se compone de dos partes, **Parte A** y **Parte B**, cada una con un peso del **50 % dentro del trabajo de investigación**.

Parte A. Investigación Matemática (Video)

- **Objetivo:** Explicar los conceptos fundamentales relacionados con las funciones. Para ello, deberán realizar un video en el que participen todos los integrantes del grupo. En el video deberán:
 - Explicar la definición de función, con énfasis en funciones polinómicas y racionales.
 - Describir las formas de representación gráfica, identificando claramente las variables independiente y dependiente.
 - Evaluar un punto en cada función e interpretar el resultado obtenido.
 - Explicar los conceptos de dominio y recorrido, ejemplificándolos con un gráfico elaborado en GeoGebra de una función racional.
 - Identificar las intersecciones con los ejes, interpretando su significado directamente en el gráfico.
- **Producto:** Un video de 4 a 5 minutos en grupo de 4 integrantes.

Parte B. Proyecto en Python (Defensa presencial)

- **Objetivo:** Implementar un programa en Python que funcione como un analizador de funciones con **interfaz gráfica**. La aplicación debe permitir al usuario ingresar una función matemática y, opcionalmente, un valor x para evaluar, es decir, dos campos de entrada. El programa deberá analizar la función para calcular y mostrar su **dominio, recorrido e intersecciones con los ejes**, exponiendo además el desarrollo computacional que justifica dichos resultados.

Al evaluar un punto, el programa deberá mostrar de forma detallada el **paso a paso de la sustitución y cálculo de $f(x)$** , concluyendo con la presentación del par ordenado resultante. Posteriormente, se debe generar una **gráfica** clara y profesional (con títulos y etiquetas) que represente la función y sus intersecciones, **resaltando el punto evaluado en un color distinto sin alterar la gráfica original**.

Se espera que el programa cuente con una **interfaz de usuario intuitiva** y un robusto **manejo de errores**.

- **Producto:** Código fuente en Python, alojado en un repositorio de **GitHub** aplicando una estructura de proyecto organizada.
 - Se **prohíbe** el uso de la librería numpy o similares.
 - Se **recomienda** utilizar librerías como Matplotlib para los graficos, sympy para el parser/entrada limpia de las respectivas funciones.
- **Defensa:**
 - Las defensas se realizarán los días **lunes 15 y martes 16 de septiembre**.
 - Cada grupo dispondrá de **5 a 6 minutos** para explicar su código y su funcionamiento.
 - Posteriormente habrá una **ronda de preguntas**, alcanzando un máximo de 10 minutos por grupo.
 - Las defensas se organizarán en tres bloques:
 1. Lunes 15 de 08:00 a 10:00 hrs.
 2. Lunes 15 de 15:00 a 17:00 hrs.
 3. Martes 16 de 15:00 a 17:00 hrs
 - El día 15 de septiembre no se realizará la clase en horario regular, pues en ese periodo habrá defensa.
 - Los ayudantes del curso participarán formulando preguntas técnicas.
- **Evaluación:** Se aplicará la rúbrica de Python diseñada por los ayudantes (ver sección correspondiente).

4. Procedimiento y entrega

1. **Conformación de grupos:** 4 estudiantes.
2. **Entregas:**

- Video (formato MP4 o link privado de YouTube).
 - Archivo en Python: `.py`.
3. **Plazo de entrega del video y código:** Sábado 13 de septiembre hasta las 23:59 hrs, vía plataforma institucional.
 4. **Importante:** Trabajos fuera de plazo tendrán como nota máxima 4,0.
 5. **Plagio y uso de IA:** Se revisará la originalidad.

5. Bibliografía recomendada

Bibliografía obligatoria

- Baldor, A. (2007). *Álgebra*. México: Grupo Editorial Patria.
- Swokowski, E. W. (2006). *Álgebra y trigonometría con geometría analítica*. México: Cengage Learning Editores.
- Zill, D. G. (2000). *Álgebra y trigonometría*. Bogotá: McGraw-Hill Interamericana.

Bibliografía complementaria

- Lehmann, Ch. (1996). *Álgebra*. Limusa.
- Sullivan, M. (1996). *Precalculus*. Prentice-Hall International.
- Vance, E. (1990). *Introducción a la matemática moderna*. Addison-Wesley Iberoamericana.
- Vance, E. (1978). *Álgebra y trigonometría*. Fondo Educativo Interamericano.

Rúbrica Parte A: Investigación Matemática (Video)

Criterio	Deficiente	Aceptable	Bueno	Excelente
1. Definición y conceptos				
Define qué es una función con claridad, indicando la relación entre variable independiente (x) y dependiente (y).	<input type="checkbox"/> (0)	<input type="checkbox"/> (1)	<input type="checkbox"/> (2)	<input type="checkbox"/> (3)
Explica qué son las funciones polinómicas.	<input type="checkbox"/> (0)	<input type="checkbox"/> (1)	<input type="checkbox"/> (2)	
Explica qué son las funciones racionales.	<input type="checkbox"/> (0)	<input type="checkbox"/> (1)	<input type="checkbox"/> (2)	
2. Ejemplos y gráficas				
Presenta al menos un ejemplo de gráfica en GeoGebra (u otro software).	<input type="checkbox"/> (0)	<input type="checkbox"/> (1)	<input type="checkbox"/> (2)	<input type="checkbox"/> (3)
Muestra cómo se evalúa una función en un punto e interpreta la coordenada (x, y) en el gráfico.	<input type="checkbox"/> (0–1)	<input type="checkbox"/> (2)	<input type="checkbox"/> (3)	<input type="checkbox"/> (4)
3. Dominio y recorrido				
Explica el concepto de dominio con un ejemplo.	<input type="checkbox"/> (0)	<input type="checkbox"/> (1)	<input type="checkbox"/> (2)	<input type="checkbox"/> (3)
Explica las restricciones del dominio en funciones racionales.	<input type="checkbox"/> (0)	<input type="checkbox"/> (1)	<input type="checkbox"/> (2)	<input type="checkbox"/> (3)
Explica el concepto de recorrido con un ejemplo.	<input type="checkbox"/> (0)	<input type="checkbox"/> (1)	<input type="checkbox"/> (2)	<input type="checkbox"/> (3)
4. Intersecciones				
Determina e interpreta las intersecciones de una función con los ejes.	<input type="checkbox"/> (0–1)	<input type="checkbox"/> (2)	<input type="checkbox"/> (3)	<input type="checkbox"/> (4)
5. Presentación				
Claridad de la exposición (lenguaje técnico adecuado, orden de ideas, duración entre 4 y 5 minutos).	<input type="checkbox"/> (0)	<input type="checkbox"/> (1)	<input type="checkbox"/> (2)	<input type="checkbox"/> (3)
Participación equitativa de todos los integrantes en el video.	<input type="checkbox"/> (0)	<input type="checkbox"/> (1)	<input type="checkbox"/> (2)	

Puntajes de referencia:

- Puntaje ideal (nota 7.0): 32 puntos.
- Puntaje de aprobación (nota 4.0): 19 puntos (60 % de exigencia).

Rúbrica Parte B: Aplicación informática en Python

- La evaluación constará de **8 preguntas para cada grupo**, las cuales estarán relacionadas con los criterios de la siguiente rúbrica. Cada grupo deberá responder en base al desarrollo de su proyecto en Python.
- Cada grupo deberá presentar su código de forma cronometrada explicando su funcionamiento, y al finalizar cada integrante responderá dos preguntas técnicas relacionadas con las decisiones de implementación para comprobar su comprensión.

Criterio	Deficiente	Aceptable	Bueno	Excelente
1. Funcionalidad Matemática				
Presenta un código en Python para evaluar funciones.	<input type="checkbox"/> (0 pts)	<input type="checkbox"/> (1 pt)	<input type="checkbox"/> (2 pts)	<input type="checkbox"/> (3 pts)
El código determina el dominio de la función ingresada.	<input type="checkbox"/> (0 pts)	<input type="checkbox"/> (1 pt)	<input type="checkbox"/> (2 pts)	<input type="checkbox"/> (3 pts)
El código determina el recorrido de la función ingresada.	<input type="checkbox"/> (0 pts)	<input type="checkbox"/> (1 pt)	<input type="checkbox"/> (2 pts)	<input type="checkbox"/> (3 pts)
El código calcula y muestra las intersecciones con los ejes .	<input type="checkbox"/> (0 pts)	<input type="checkbox"/> (1 pt)	<input type="checkbox"/> (2 pts)	<input type="checkbox"/> (3 pts)
La aplicación expone el desarrollo paso por paso que justifica los resultados obtenidos (dominio, recorrido, etc.).	<input type="checkbox"/> (0 pts)	<input type="checkbox"/> (1 pt)	<input type="checkbox"/> (2 pts)	<input type="checkbox"/> (3 pts)
Permite evaluar un punto x en la función y remarca el punto resultante en la gráfica.	<input type="checkbox"/> (0 pts)	<input type="checkbox"/> (1 pt)	<input type="checkbox"/> (2 pts)	<input type="checkbox"/> (3 pts)
2. Gráficas				
El gráfico incluye el dominio visible , el recorrido visible y todas las intersecciones con los ejes (este contiene títulos, etiquetas en ejes, leyenda clara y colores).	<input type="checkbox"/> (0-1 pts)	<input type="checkbox"/> (2-3 pts)	<input type="checkbox"/> (4-5 pts)	<input type="checkbox"/> (6-8 pts)
3. Uso de librerías simbólicas				
El ingreso de funciones mediante <code>input()</code> o similar debe ingresarse de la manera más limpia posible (por ejemplo, $2x^2 + 2x$). Se puede utilizar una librería simbólica (Sympy u otra similar).	<input type="checkbox"/> (0 pts)	<input type="checkbox"/> (1 pt)	<input type="checkbox"/> (2 pts)	<input type="checkbox"/> (3 pts)
4. Calidad del código				
Se implementa manejo de errores (entradas inválidas, divisiones por cero, acepta fracciones, acepta decimales, etc.).	<input type="checkbox"/> (0 pts)	<input type="checkbox"/> (1 pt)	<input type="checkbox"/> (2 pts)	<input type="checkbox"/> (3 pts)

Criterio	Deficiente	Aceptable	Bueno	Excelente
El proyecto utiliza una arquitectura limpia (main.py + módulos auxiliares).	<input type="checkbox"/> (0 pts)	<input type="checkbox"/> (1 pt)	<input type="checkbox"/> (2 pts)	<input type="checkbox"/> (3 pts)
5. Interfaz				
El proyecto contiene el uso de una interfaz, esta es intuitiva y amigable con el usuario.	<input type="checkbox"/> (0-1 pts)	<input type="checkbox"/> (2-3 pts)	<input type="checkbox"/> (4-5 pts)	<input type="checkbox"/> (6-8 pts)
6. Documentación y GitHub				
El grupo contiene un repositorio en GitHub para su desarrollo con colaboradores, además de contar con un mínimo de un commit por integrante.	<input type="checkbox"/> (0 pts)	<input type="checkbox"/> (1 pt)	<input type="checkbox"/> (2 pts)	<input type="checkbox"/> (3 pts)
7. Presentación final				
El código es explicado con claridad.	<input type="checkbox"/> (0-1 pts)	<input type="checkbox"/> (2-3 pts)	<input type="checkbox"/> (4-5 pts)	<input type="checkbox"/> (6-8 pts)
Se utiliza un lenguaje técnico preciso y adecuado durante la presentación.	<input type="checkbox"/> (0-1 pts)	<input type="checkbox"/> (2-3 pts)	<input type="checkbox"/> (4-5 pts)	<input type="checkbox"/> (6-8 pts)
Se muestra una ejecución completa con ingreso de función, cálculo y gráfico.	<input type="checkbox"/> (0-1 pts)	<input type="checkbox"/> (2-3 pts)	<input type="checkbox"/> (4-5 pts)	<input type="checkbox"/> (6-8 pts)
Responde de manera correcta las preguntas planteadas	<input type="checkbox"/> (0-1 pts)	<input type="checkbox"/> (2-3 pts)	<input type="checkbox"/> (4-5 pts)	<input type="checkbox"/> (6-8 pts)
El tiempo de presentación no debe pasar de los 5-6 minutos	<input type="checkbox"/> (0-1 pts)	<input type="checkbox"/> (2-3 pts)	<input type="checkbox"/> (4-5 pts)	<input type="checkbox"/> (6-8 pts)

Puntajes de referencia:

- **Puntaje ideal (nota 7.0):** 86 puntos.
- **Puntaje de aprobación (nota 4.0):** 52 puntos (60 % de exigencia).