



Módulo 3

Sesión N° 4



ACTIVIDAD:

Optimización y Análisis Geométrico de Funciones en Dos Variables

- Objetivo: Aplicar los conceptos del cálculo diferencial multivariable para calcular derivadas parciales de una función de dos variables, determinar el gradiente y construir la matriz Hessiana.



Instrucciones:

1. Definición de la Función:

- Definir simbólicamente una función de dos variables $g(x, y)$ que presente un punto crítico identificable.
- Ejemplo sugerido:

$$g(x, y) = x^2 + 3y^2 - 4x + 2y + 1$$

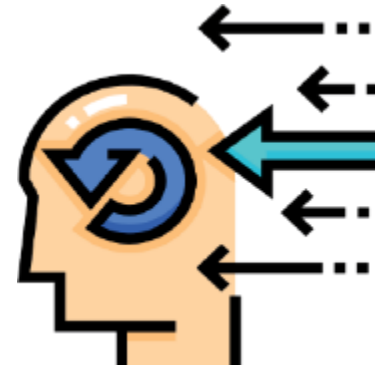
2. Cálculo de Derivadas Parciales, Gradiente y Matriz Hessiana:

- Utilizar la librería Sympy para:
 - Calcular las derivadas parciales $\frac{\partial g}{\partial x}$ y $\frac{\partial g}{\partial y}$.
 - Determinar el gradiente $\nabla g(x, y)$, es decir, el vector de derivadas parciales.
 - Construir la matriz Hessiana, que incluya las segundas derivadas parciales.
- Resolver el sistema dado por $\nabla g(x, y) = (0, 0)$ para encontrar el/los punto(s) crítico(s).



3. Clasificación del Punto Crítico:

- Evaluar la matriz Hessiana en el punto crítico encontrado y utilizar sus valores propios para clasificar el punto (mínimo local, máximo local o punto de silla).



4. Visualización:

- Convertir la función simbólica en una función numérica mediante `sp.lambdify`.
- Generar un meshgrid de valores de x e y en un rango adecuado (por ejemplo, $x \in [-5,5], y \in [-5,5]$).
- Graficar la superficie 3D de $g(x, y)$ usando Matplotlib y marcar claramente el punto crítico, destacando su clasificación.
- Complementar el gráfico con un mapa de contorno (contour plot) para visualizar la distribución de valores en el plano.

5. Relación con Machine Learning:

- Incluir una breve discusión escrita o en el código (comentarios) que explique cómo el cálculo del gradiente y la clasificación de puntos críticos es fundamental para técnicas de optimización, como el descenso de gradiente utilizado en el entrenamiento de modelos de Machine Learning.



6. Documentación y Evidencias:

- Cada bloque de código debe incluir docstrings y comentarios explicativos.
- Se debe entregar un informe breve (o README) que resuma la estructura del código, la metodología empleada y la relevancia de los conceptos desarrollados en el contexto de Machine Learning.
- Se deben incluir capturas de pantalla del código ejecutado y de los gráficos generados, evidenciando la solución y la visualización del punto crítico.

7. Entrega:

- Tiempo estimado de desarrollo: 2 horas.
- Formato de ejecución: individual.

