**Fundamentos Matemáticos**

**Unidad 1. : Razonamiento Matemático.​**

**​1.1 Proposiciones y conectivos lógicos.​**

**1.2 Tautología y contradicción.​**

**1.3 Álgebra proposicional y equivalencias.​**

**1.4 Aplicación a la teoría de circuitos y otras áreas.​**

**1.5 Razonamiento válido.​**

**1.6 Cuantificadores.​**

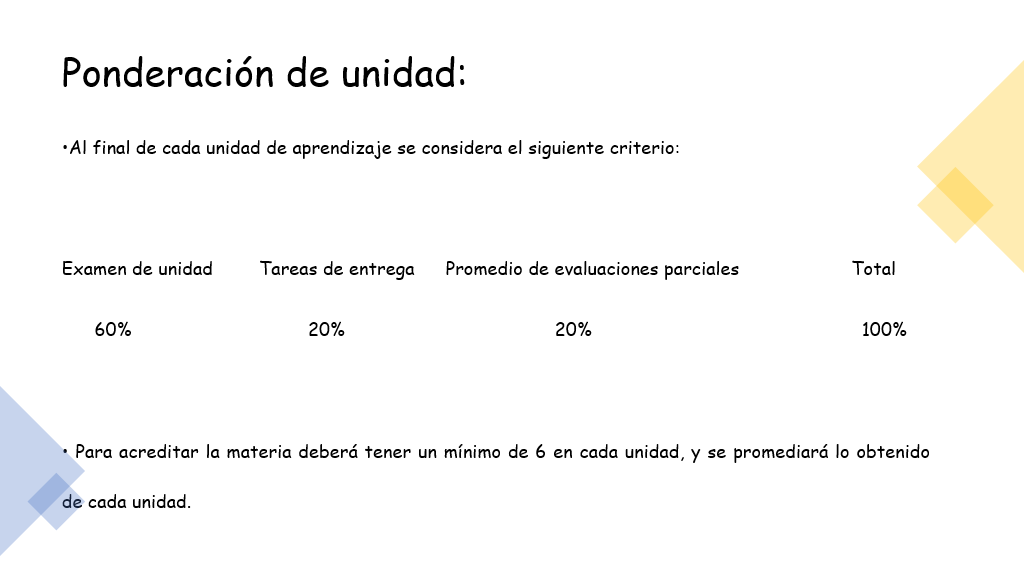
**1.7 Método deductivo de demostración.​**

**​**

**Unidad 2. Conjuntos​**

**​2.1 Definición y determinación de conjuntos.​**

**2.2 Resolución de problemas de cardinalidad.​**

****

**ALGEBRA**

  • Unidad 1: Matrices y Sistemas de Ecuaciones Lineales

• Unidad 2: Determinantes

• Unidad 3: Principio de Inducción

• Unidad 4: Números Complejos y sus propiedades

• Unidad 5: Polinomios Reales

Rúbrica

• Promedio Exámenes Parciales: 70 %

• Ejercicios en clase: 20 %

• Tareas: 10 %

• Total 100%

• Requisito: Aprobar todos los exámenes parciales

**INTRO CIENCIA DE DATOS**

Examen departamental y/o Exámenes 30%

Informe de prácticas y tareas 30%

Proyecto de ciencia de datos 40%

Total 100%

**Temas que vienen (para que estudies):**

1. **Conceptos básicos de Ciencia de Datos**
   * ¿Qué es la Ciencia de Datos?
   * Diferencias entre **IA (Inteligencia Artificial)** y **Ciencia de Datos**.
   * Rol del **Data Analyst**.
2. **Datos y sus tipos**
   * Datos estructurados, semi-estructurados y no estructurados.
   * Bases de datos (DB).
   * **Big Data** y sus 5 V’s: Volumen, Velocidad, Variedad, Veracidad y Valor.
3. **Procesos y herramientas**
   * **ETL (Extract, Transform, Load)**.
   * **Data Mining** (descubrimiento de patrones).
   * **Data Cleaning** (limpieza de datos).
   * **Machine Learning** (supervisado, no supervisado, por refuerzo).
   * Algoritmos.
   * **Data Pipeline** (flujo de datos).
   * **Clusterización**.
   * Uso de plataformas en la nube como **AWS**.
4. **Proyectos de análisis de datos**
   * Estructura de un proyecto.
   * Ejemplo: análisis de la afluencia del metro CDMX.
5. **Generaciones en ingeniería de datos**
   * **1ª Gen.**: Bases de datos relacionales (SQL).
   * **2ª Gen.**: Data Warehousing y ETL.
   * **3ª Gen.**: Big Data y NoSQL.
   * **4ª Gen.**: Data Lakes y Cloud Computing.
   * **5ª Gen.**: DataOps y MLOps.
6. **Modelos y tareas en Ciencia de Datos**
   * **Modelos descriptivos** (patrones, resúmenes).
   * **Modelos predictivos** (predicciones futuras).

**Tareas descriptivas:**

* + Agrupamiento (clustering).
  + Correlaciones y factorizaciones.
  + Reglas de asociación.
  + Dependencias funcionales.

**Tareas predictivas:**

* + Clasificación.
  + Clasificación suave.
  + Categorización.
  + Priorización.
  + Regresión.

✅ **Lo esencial que te recomiendo estudiar primero:**

* Qué es Ciencia de Datos y en qué se diferencia de IA.
* Tipos de datos y Big Data.
* Procesos clave: ETL, Data Mining, Data Cleaning, Machine Learning.
* Generaciones de la ingeniería de datos.
* Modelos descriptivos vs predictivos y ejemplos de sus tareas

UNIDAD 2

📘 **Unidad 2 – Ciclo de vida del análisis de datos**

**Conceptos básicos**

* Los proyectos de ciencia de datos son más **exploratorios** que los de inteligencia de negocios.

**Roles en análisis de datos**

* **Business User** → experto en el dominio del negocio.
* **Project Sponsor** → impulsa el proyecto, define el problema y financia.
* **Project Manager** → gestiona tiempos, recursos y entregables.
* **Business Intelligence Analyst** → genera reportes y dashboards.
* **Database Administrator (DBA)** → administra las bases de datos.
* **Data Engineer** → construye pipelines y maneja datos.
* **Data Scientist** → crea modelos analíticos.

**Herramientas clave**

* **Apache Spark** → procesamiento masivo de datos (lotes y tiempo real).
* **Dashboards y visualización** → Matplotlib, Seaborn, Plotly, Dash, Power BI, Tableau.

**Fases del ciclo de vida del análisis de datos**

1. **Descubrimiento (Discovery)**
   * Aprender el dominio del negocio.
   * Evaluar recursos disponibles.
   * Definir el problema (**Framing the Problem**).
   * Identificar stakeholders clave.
   * Formular hipótesis iniciales.
   * Identificar y evaluar fuentes de datos.

📌 Ejemplo: *Bank churn (abandono de clientes en un banco)* → hipótesis sobre edad, saldo, satisfacción, etc.

1. **Preparación de datos (ETL)**
   * **Extracción**: CSV, SQL, APIs, Excel, JSON, XML, Web Scraping, Hadoop.
   * **Transformación (Data Cleaning)**:
     + Eliminar valores nulos/duplicados.
     + Tratar outliers.
     + Corregir formatos y errores.
     + Normalización y escalado.
     + Imputación de valores faltantes.
   * **Carga**: Bases SQL, Data Warehouses (BigQuery, Snowflake, Redshift), archivos (CSV, JSON).
2. **Planeación y selección del modelo**
   * Según el sector:
     + Bancos → regresión múltiple.
     + Telecomunicaciones → redes neuronales.
     + Consumo → regresión lineal múltiple.
   * **Modelos candidatos**: clasificación, regresión, clustering, etc.
3. **Modelado**
   * **Aprendizaje supervisado**
     + Clasificación: KNN, Árboles de decisión, SVM, Naive Bayes, Redes Neuronales.
     + Regresión: lineal, polinómica, logística.
     + Métricas: Matriz de confusión, precisión, recall, F1-score, ROC, MSE, MAE, R².
   * **Aprendizaje no supervisado**
     + Clusterización: K-Means, DBSCAN, Clustering jerárquico, GMM.
     + Reducción de dimensionalidad: PCA, t-SNE.
     + Métricas: Silhouette, Inercia, Davies-Bouldin.
   * **Aprendizaje por refuerzo (Reinforcement Learning)**
     + Q-Learning, SARSA.
     + Métodos: Monte Carlo, Temporal Difference (TD).
4. **Comunicación de resultados**
   * Dashboards y reportes para distintos públicos (ejecutivos, analistas, técnicos).
   * Visualizaciones limpias y claras.
5. **Operacionalización**
   * Implementación en piloto antes de escalar.
   * Documentación técnica.
   * Presentaciones para sponsors, analistas y equipo técnico.

✅ **Lo esencial para tu examen:**

* Roles del equipo de ciencia de datos.
* Fases del ciclo de vida (Discovery → Preparación → Modelado → Comunicación → Operacionalización).
* Herramientas ETL y limpieza de datos.
* Tipos de aprendizaje (supervisado, no supervisado, por refuerzo).
* Algoritmos principales de clasificación, regresión y clustering.
* Métricas de evaluación (precisión, recall, F1, ROC, MSE, MAE, R²).

**PROGRAMACION BASICA**

**📘 Unidad 1 – Introducción a la programación estructurada**

**Temas principales para estudiar**

1. **Paradigmas de la programación**
   * **Diferentes formas de pensar y organizar programas.**
   * **Ejemplos: programación estructurada, orientada a objetos, funcional, lógica.**
2. **Programación estructurada**
   * **Base de la unidad.**
   * **Uso de estructuras de control (secuencia, selección, repetición).**
   * **Enfoque en dividir los programas en bloques claros y ordenados.**
3. **Variables**
   * **Espacios de memoria donde se almacenan datos.**
   * **Tipos de datos (entero, real, carácter, booleano).**
   * **Declaración y uso de variables en un programa.**

**⚡ Como el archivo está muy reducido (apenas tiene título y unas diapositivas), parece que es solo la presentación inicial de la materia, así que lo más seguro es que en clase desarrollen:**

* **Tipos de datos.**
* **Operadores (aritméticos, relacionales, lógicos).**
* **Estructuras de control (if, while, for).**
* **Diagramas de flujo y pseudocódigo.**

**✅ Lo esencial para repasar ahora mismo:**

* **Paradigma de la programación estructurada.**
* **Qué es una variable y cómo se declara.**
* **Tipos de datos básicos.**
* **Principio de dividir un programa en pasos lógicos (inicio → proceso → fin).**

**1. Tipos de variables**

* **str: cadenas de texto.**
* **int: números enteros.**
* **float: números decimales.**
* **bool: valores lógicos (True, False).**

**Ejemplo:**

**nombre = "Juan" # str**

**edad = 21 # int**

**altura = 1.75 # float**

**es\_estudiante = True # bool**

**Actividad:  
Declara una variable de cada tipo y muéstrala con print().**

**2. Listas (list)**

**Estructura que almacena varios valores. Son mutables.**

**Ejemplo:**

**numeros = [1, 2, 3, 4, 5]**

**nombres = ["Ana", "Luis", "Marta"]**

**mixta = [10, "Python", 3.14, True]**

**print(nombres[2]) # "Marta"**

**Actividad:  
Crea una lista con tus 3 materias favoritas y muestra la segunda.**

**3. Operadores aritméticos**

**Permiten hacer cálculos.**

* **+ suma**
* **- resta**
* **\* multiplicación**
* **/ división**
* **\*\* potencia**
* **% módulo (resto)**

**Ejemplo:**

**a, b = 6, 8**

**print(a+b, a-b, a\*b, a/b, a\*\*b, a%b)**

**Actividad:  
Pide dos números al usuario y muestra todas las operaciones.**

**4. Entrada de datos: input()**

**Recibe datos del usuario en formato cadena (str).  
Se convierte con int() o float() si es necesario.**

**Ejemplo:**

**edad = int(input("¿Cuántos años tienes? "))**

**print("Tienes", edad, "años")**

**Actividad:  
Haz un programa que pida tu nombre y edad, y luego diga:  
Hola Juan, tienes 20 años.**

**5. Formateo de cadenas: f-strings**

**Permiten insertar variables directamente en el texto.**

**Ejemplo:**

**nombre, edad = "Juan", 21**

**print(f"Hola, {nombre}! Tienes {edad} años.")**

**Actividad:  
Haz un programa que pida tu nombre y tu estatura, y muestre:  
Soy Ana y mido 1.68 metros.**

**6. type()**

**Muestra el tipo de dato de una variable.**

**Ejemplo:**

**x = 3.5**

**print(type(x)) # <class 'float'>**

**Actividad:  
Declara 4 variables diferentes (str, int, float, bool) y usa type() para mostrar sus tipos.**

**7. Diccionarios (dict)**

**Estructura que guarda datos en pares clave-valor.**

**Ejemplo:**

**estudiante = {"nombre": "Juan", "edad": 21, "promedio": 9.0}**

**print(estudiante["nombre"]) # "Juan"**

**Actividad:  
Crea un diccionario con tu nombre, edad y color favorito. Imprímelo usando un f-string.**

**8. Operaciones con listas**

* **append(x) → agrega al final.**
* **insert(pos, x) → inserta en posición.**
* **+= [x,y] → agrega múltiples.**
* **remove(x) → elimina el valor.**
* **pop(pos) → elimina por posición.**
* **del lista[i] → borra un elemento.**

**Ejemplo:**

**mi\_lista = []**

**mi\_lista.append(10)**

**mi\_lista.insert(0, 5)**

**print(mi\_lista) # [5, 10]**

**Actividad:  
Crea una lista de números, agrega un nuevo valor con append() y luego elimina el primero con pop().**

**9. Operaciones con diccionarios**

* **Agregar o modificar valores: dic["clave"] = valor**
* **Eliminar valores: pop("clave") o del dic["clave"]**

**Ejemplo:**

**dic = {"nombre": "Ana", "edad": 20}**

**dic["edad"] = 21**

**dic["color"] = "azul"**

**print(dic)**

**dic.pop("color")**

**Actividad:  
Crea un diccionario con tu nombre y comida favorita. Luego cámbiala por otra y elimina el nombre.**

**10. Funciones**

**Bloques de código reutilizables.**

* **Definir: def nombre\_funcion():**
* **Parámetros: reciben valores.**
* **return: devuelve resultados.**

**Ejemplo básico:**

**def saludar(nombre):**

**print(f"Hola, {nombre}!")**

**saludar("Juan")**

**Ejemplo con retorno:**

**def suma(a, b):**

**return a + b**

**print(suma(3, 5)) # 8**

**Actividad:  
Haz una función llamada area\_circulo(r) que reciba el radio y calcule el área (π\*r^2).**

**1. print()**

* **Definición: Muestra texto o valores en consola.**
* **Ejemplo:**

**print("Hola estudio ciencia de datos")**

**print("Mi nombre es Ana")**

* **Ejercicio: Escribe un programa que muestre tu nombre, edad y carrera usando print().**

**2. Tipos de datos (int, str, float, bool)**

* **Definición: Tipos básicos en Python:**
  + **int → números enteros**
  + **str → cadenas de texto**
  + **float → números decimales**
  + **bool → valores lógicos (True/False)**
* **Ejemplo:**

**a = 3 # int**

**b = "Hola" # str**

**c = 12.43 # float**

**d = True # bool**

* **Ejercicio: Declara 4 variables (una de cada tipo) y muéstralas con print().**

**3. type()**

* **Definición: Devuelve el tipo de dato de una variable.**
* **Ejemplo:**

**x = 3.14**

**print(type(x)) # <class 'float'>**

* **Ejercicio: Crea una variable de cada tipo y usa type() para comprobar su tipo.**

**4. Conversión de tipos (int(), str(), float())**

* **Definición: Permiten transformar un valor de un tipo a otro.**
* **Ejemplo:**

**a = "3"**

**print(int(a) + 2) # 5**

**print(str(3) + "2") # "32"**

* **Ejercicio: Pide un número como texto y conviértelo a entero para sumarle 10.**

**5. Operadores aritméticos**

* **Definición: Permiten hacer operaciones matemáticas.**
  + **+ suma**
  + **- resta**
  + **\* multiplicación**
  + **/ división**
  + **// división entera**
  + **% módulo (resto)**
  + **\*\* potencia**
* **Ejemplo:**

**a = 5**

**b = 3**

**print(a+b, a-b, a\*b, a/b, a//b, a%b, a\*\*b)**

* **Ejercicio: Haz un programa que pida 2 números y muestre todas las operaciones.**

**6. Operadores relacionales**

* **Definición: Comparan dos valores y devuelven True o False.**
* **Ejemplo:**

**a = 5**

**b = 3**

**print(a == b) # False**

**print(a > b) # True**

* **Ejercicio: Pide dos números y compara si son iguales, mayores o menores.**

**7. Operadores lógicos (and, or, not)**

* **Definición: Permiten combinar condiciones lógicas.**
* **Ejemplo:**

**a = True**

**b = False**

**print(a and b) # False**

**print(a or b) # True**

**print(not a) # False**

* **Ejercicio: Haz un programa que pida edad y si tiene licencia, y diga si puede conducir (edad >= 18 and licencia == True).**

**8. Condicional if**

* **Definición: Ejecuta un bloque de código si la condición es verdadera.**
* **Ejemplo:**

**edad = 17**

**if edad >= 18:**

**print("Puede entrar")**

* **Ejercicio: Pide un número y di si es positivo.**

**9. Condicional if-else**

* **Definición: Ejecuta un bloque si la condición es verdadera, y otro si es falsa.**
* **Ejemplo:**

**temp = -10**

**if temp > 30:**

**print("Día caluroso")**

**else:**

**print("Día fresco")**

* **Ejercicio: Pide un número y muestra si es par o impar.**

**10. Condicional if-elif-else**

* **Definición: Permite evaluar varias condiciones en orden.**
* **Ejemplo:**

**hora = 23**

**if hora < 12:**

**print("Buenos días")**

**elif hora < 19:**

**print("Buenas tardes")**

**else:**

**print("Buenas noches")**

* **Ejercicio: Pide una calificación (0–100) y muestra:**
  + **“Reprobado” si es <60**
  + **“Regular” si es 60–79**
  + **“Bien” si es 80–89**
  + **“Excelente” si es 90–100**

**✅ Actividad final integradora**

**Crea un programa que:**

1. **Pida el nombre y edad del usuario.**
2. **Solicite 2 números.**
3. **Guarde en una lista la suma, resta, multiplicación, división y potencia.**
4. **Muestre los resultados en un diccionario con claves "suma", "resta", etc.**
5. **Presente todo con un f-string.**