**PROPUESTAS PROYECTO TERCER PARCIAL**

**PRIMERA OPCIÓN**

* **Simulador de Estrategias para Juegos de Mesa (Ajedrez, Sudoku, etc.)**

**Descripción:**  
Un sistema que prueba diferentes estrategias para juegos de mesa y encuentra la mejor jugada.

**Técnicas utilizadas:**

* **Listas**: Para representar el tablero.
* **Árboles de decisión**: Para calcular las mejores jugadas.
* **Ordenamientos**: Para priorizar los movimientos óptimos.
* **Backtracking**: Para explorar todas las posibles jugadas.
* **Algoritmos voraces**: Para elegir la mejor jugada en cada turno.
* **Programación dinámica**: Para evaluar estrategias a largo plazo.
* **Fuerza bruta**: Para simular todas las combinaciones de jugadas.

**Salida:**

* **MATLAB:** Análisis de estrategias óptimas.
* **Power BI:** Gráficos con estadísticas de victorias y eficiencia de jugadas.

### ****📌 Simulador de Juegos de Mesa**** 🎲

**🔹 Ventajas:**  
✔️ Más llamativo y divertido para la presentación.  
✔️ Impresiona porque usa **árboles de decisión**, **backtracking** y **algoritmos voraces**.  
✔️ Puedes implementar varios juegos (ajedrez, sudoku, damas).  
✔️ La simulación de estrategias da buenos gráficos en MATLAB y Power BI.

**⚠️ Desafíos:**  
❌ Más difícil de programar.  
❌ Dependiendo del juego, puede requerir inteligencia artificial básica.

### SEGUNDA OPCIÓN

### ****Planificador Inteligente de Horarios Universitarios****

**Técnicas utilizadas:**

* **Listas enlazadas:** Para almacenar materias y horarios.
* **Árbol de decisión (Decision Tree):** Para elegir la mejor combinación de clases.
* **Ordenamientos:** Aplicar **HeapSort** para priorizar clases con mayor demanda.
* **Backtracking:** Para probar todas las combinaciones de horarios.
* **Programación Dinámica:** Para optimizar la cantidad de clases por día.
* **Fuerza Bruta:** Para comparar todas las combinaciones posibles.

**Salida:**

* **MATLAB:** Generar gráficos de ocupación de aulas por día.
* **Power BI:** Exportar informes con disponibilidad de horarios.

### ****📌 Planificador Inteligente de Horarios**** 🏫

**🔹 Ventajas:**  
✔️ Más fácil de implementar con listas, árboles y programación dinámica.  
✔️ Útil en la vida real, más fácil de justificar en la entrega.  
✔️ Requiere menos optimización matemática que el simulador de juegos.  
✔️ Puede exportar fácilmente datos a Power BI para mostrar análisis de horarios.

**⚠️ Desafíos:**  
❌ Menos visualmente impresionante que un simulador de juegos.  
❌ Requiere lógica para manejar conflictos de horarios y restricciones.

### TERCERA OPCIÓN

### ****Simulación de Bolsa de Valores****

**Descripción:**  
Un sistema que simula la compra y venta de acciones con predicción de precios.

**Técnicas utilizadas:**

* **Listas**: Para almacenar precios históricos.
* **Árboles AVL**: Para buscar valores de acciones rápidamente.
* **Ordenamientos (MergeSort, QuickSort)**: Para ordenar precios históricos.
* **Algoritmos voraces**: Para hacer inversiones rápidas.
* **Backtracking**: Para probar estrategias de inversión.
* **Programación dinámica**: Para encontrar la mejor estrategia a largo plazo.
* **Fuerza bruta**: Para comparar todas las estrategias posibles.

**Salida:**

* **MATLAB:** Gráficos de tendencias de acciones.
* **Power BI:** Análisis de rendimiento de inversión.

### CUARTA OPCIÓN

### ****Sistema de Predicción del Clima****

**Descripción:**  
Un sistema que usa datos históricos para predecir el clima de una región.

**Técnicas utilizadas:**

* **Listas**: Para almacenar datos climáticos.
* **Árboles AVL**: Para búsqueda rápida de datos.
* **Ordenamientos (MergeSort, QuickSort)**: Para organizar temperaturas.
* **Backtracking**: Para analizar tendencias meteorológicas.
* **Algoritmos voraces**: Para hacer predicciones rápidas.
* **Programación dinámica**: Para optimizar modelos climáticos.
* **Fuerza bruta**: Para comparar predicciones con datos reales.

**Salida:**

* **MATLAB:** Gráficos de tendencias meteorológicas.
* **Power BI:** Comparación entre predicción y clima real.