

Problema 1:

Desarrollar el motor lógico de un cajero automático que resuelva la entrega de efectivo mediante Programación Dinámica, gestionando el inventario y diferenciando los mensajes de usuario de las alertas técnicas de mantenimiento.

No solo debes calcular cuántos billetes se entregan, sino cuáles.

Antes de confirmar una transacción, el sistema debe verificar si hay billetes suficientes en el inventario físico para cubrir la solución óptima encontrada.

Acciones para el usuario: El usuario solo debe ver información relevante para su experiencia:

- Si el usuario pide un monto menor al billete más pequeño disponible en el stock actual, el sistema debe indicar el rechazo de la operación y cual es el monto mínimo a extraer..
- Si la operación es exitosa, indicar el monto que se entregará indicando la cantidad de billetes de cada denominación que el usuario recibirá
- Si el cajero no puede formar el monto con los billetes disponibles o si el inventario total está en cero, mostrar: "Cajero fuera de servicio o fondos insuficientes para este monto".

Acciones para el administrador del sistema:

- Alerta de reabastecimiento: Se debe disparar una alerta cuando una denominación específica caiga por debajo de un umbral mínimo (ejemplo: menos de 10 unidades).

Problema 2:

Una ciudad consta de 8 distritos estratégicos. El gobierno necesita conectar todos los distritos con una red de fibra óptica subterránea, pero con la condición de minimizar el impacto ambiental, solo se pueden realizar excavaciones siguiendo senderos preexistentes. Cada sendero entre dos distritos tiene un costo de instalación asociado (basado en la distancia y la dureza del terreno). Tu objetivo es encontrar la red que conecte todos los distritos entre sí (de forma que se pueda viajar de cualquier distrito a cualquier otro) con el **costo total mínimo**.

Para resolver este problema, realizar mediante dos funciones alternativas: una donde aplique el Algoritmo de Prim y otro el de Kruskal

Requerimientos del Desarrollo

Representación del Grafo: El sistema debe recibir los distritos y sus conexiones mediante una Matriz de Adyacencia o una Lista de Adyacencia.

Visualización del Proceso: El algoritmo debe imprimir paso a paso qué conexión decide añadir y cuál es el costo acumulado en ese momento.

Resultado Final: Al terminar, el programa debe mostrar: La lista de todas las conexiones elegidas y el Costo Total de la infraestructura.

Problema 3:

Eres el programador de una empresa de mensajería. Tienes un almacén central y varios puntos de entrega dispersos . La misión es determinar el camino más rápido (menor costo/tiempo) desde el almacén hacia cada uno de los destinos.

Realizar un programa, que mediante el uso del algoritmo de Dijkstra se encuentre el camino mas corto.