

Universidad Nacional Autónoma de México

Instituto de Investigaciones de Matemáticas Aplicadas y Sistemas

Matemáticas Discretas

Sistema Automatizado de Asignación de Capacitación

Proyecto Final – 5to Semestre

Integrantes

Abril Montaña

Emiliano Mendoza

José Vera

Introducción

Contexto del Proyecto

La Encuesta de Necesidades de Capacitación (ENC) es una herramienta que usa el INE para saber qué habilidades necesitan mejorar los trabajadores del servicio profesional electoral. La idea es que si ellos están bien preparados, el servicio que se da a las personas será mucho mejor.

Con esta encuesta no solo se ve qué saben hacer los trabajadores, también se mide cuánta gente necesita cada curso. Eso ayuda a planear cómo cubrir esas necesidades sin desperdiciar recursos. Por lo que es necesario analizar y procesar esta información para organizar a los trabajadores en grupos y que puedan capacitarse de manera eficiente.

Objetivo General

El objetivo de este proyecto es hacer un algoritmo que asigne automáticamente a los trabajadores a los cursos que necesitan. Este algoritmo debe considerar cosas como la demanda, los cupos disponibles y algunas reglas que deben cumplirse (como priorizar a los trabajadores con más antigüedad). Con esto, se espera procesar los datos rápido y ayudar a tomar decisiones más claras para los programas de capacitación.

Objetivos Específicos

1. Crear un algoritmo que use los datos de los trabajadores y de los cursos para hacer asignaciones lógicas.
2. Hacer grupos que tengan un tamaño fijo y que no se pasen de los cupos permitidos.
3. Usar criterios como la antigüedad para que los trabajadores que más lo necesitan tengan prioridad.
4. Detectar si hay trabajadores que no fueron asignados a ningún curso y buscar maneras de resolver esto.
5. Organizar los datos para que se puedan analizar después y encontrar tendencias o patrones de demanda.

Alcance

El proyecto hará lo siguiente:

- Procesará los datos de la ENC para asignar a los trabajadores y organizarlos en grupos.
- Ayudará a identificar qué cursos no cubrieron toda la demanda.
- Generará resultados claros para tomar decisiones sobre la capacitación.

No incluye:

- Interpretar los resultados finales desde un enfoque humano o psicológico.
 - Crear nuevos cursos ni modificar los que ya existen.
 - Validar con los trabajadores si las asignaciones cumplen sus expectativas.
-

Planteamiento del Problema

Descripción del Problema

El INE tiene mucha información sobre los cursos que se necesitan en diferentes estados y modalidades, pero no hay un sistema que asigne a los trabajadores de manera ordenada. Ahora mismo no se sabe cómo:

1. Calcular cuántos grupos hacen falta para cada curso.
2. Asignar a los trabajadores según su antigüedad, empezando con los que llevan más tiempo.
3. Respetar los tamaños de los grupos y evitar que queden vacíos o saturados.

Si no se resuelve esto, algunos cursos podrían tener muy pocos participantes y otros demasiados. Esto genera problemas como desperdicio de recursos y capacitación ineficiente.

Impacto del Problema

1. **En los trabajadores:** Algunos no reciben los cursos que necesitan para mejorar su desempeño.
2. **En los cursos:** Recursos mal utilizados porque algunos cursos no tienen suficientes personas y otros están saturados.
3. **En el programa de capacitación:** Todo se vuelve más difícil de organizar y se podrían atrasar las capacitaciones.

Preguntas de Investigación

1. ¿Cómo calcular el número de grupos y modalidades necesarias para que todos los trabajadores tengan acceso a los cursos?
 2. ¿Cómo priorizar a los trabajadores más antiguos en las asignaciones?
 3. ¿Qué hacer cuando sobran o faltan participantes para que ningún curso quede incompleto?
 4. ¿Cómo manejar a los trabajadores no asignados y darles solución en el futuro?
 5. ¿Cómo asegurarse de que este algoritmo pueda usarse en futuros programas de capacitación?
3. Modelado Matemático y Justificación

Suposiciones Iniciales

Nuestra hipótesis está basada en la falta de organización de las bases de datos del INE. Durante el análisis de los datos, nos dimos cuenta de que las cifras relacionadas con las personas interesadas en cursos no siempre estaban claras. Algunas columnas tenían valores inconsistentes, faltantes o incluso duplicados. Esto nos llevó a asumir que para realizar cualquier asignación precisa, primero debíamos limpiar y estructurar los datos. Fue complicado porque de todas las bases revisadas pocas tenían una estructura utilizable para fines de este proyecto.

Definición del Modelo Matemático

1. Trabajadores

Cada trabajador se modela como un nodo T_i con las siguientes características:

- a_i : Antigüedad del trabajador T_i , representada como un valor entero positivo.
- m_i : Modalidad preferida del trabajador T_i , donde $m_i \in 1, 2$, siendo 1 para "presencial" y 2 para "a distancia".
- c_i : Curso preferido del trabajador T_i , donde $c_i \in C$, siendo C el conjunto de todos los cursos disponibles.

2. Cursos

Cada curso se modela como un conjunto C_j , definido por:

- q_j : Cupo máximo del curso C_j , un número entero positivo.
- d_j : Demanda observada del curso C_j , es decir, el número total de trabajadores que lo solicitaron, con $d_j \geq 0$.

3. Asignaciones

La relación entre trabajadores y cursos se representa mediante una función de asignación:

$$f: T \rightarrow C$$

Donde:

$$f(T_i) = C_j \quad \text{si el trabajador } T_i \text{ es asignado al curso } C_j.$$

Si un trabajador T_i no puede ser asignado a ningún curso, se define:

$$f(T_i) = -1.$$

4. Restricciones

- Cupo Máximo:

El número de trabajadores asignados a un curso C_j no puede exceder su cupo máximo q_j .

Esto se representa como:

$$\forall C_j \in C, \quad \sum_{T_i \in C_j} 1 \leq q_j.$$

5. Prioridad por Antigüedad:

Un trabajador T_i con mayor antigüedad a_i tiene prioridad sobre un trabajador T_k con menor antigüedad a_k para ser asignado al mismo curso C_j . Esto se representa como:

Si $a_i > a_k$, entonces $f(T_i) \neq -1$ antes que $f(T_k) \neq -1$.

Estas restricciones aseguran que la asignación sea eficiente y respete las prioridades establecidas en el problema.

Justificación del Modelado

Este modelo es adecuado para resolver el problema porque:

1. Datos Discretos:

Los trabajadores, los cursos y las relaciones entre ellos se representan como conjuntos finitos. Esto permite modelar el problema utilizando estructuras discretas (nodos, funciones y conjuntos), lo cual simplifica su implementación y análisis.

2. Priorización por Antigüedad:

La regla de priorizar a los trabajadores con mayor antigüedad se implementa de manera directa a través del ordenamiento de datos y las restricciones. Esto asegura que las decisiones sean consistentes con las reglas establecidas por el negocio.

3. Limitaciones del Mundo Real:

El modelo incluye funciones y restricciones que simulan escenarios reales como:

- Cupos máximos en los cursos.
- Compatibilidad con modalidades específicas.

Enunciado Formal del Problema Algorítmico

Definición Formal

Inputs:

- Una lista de n trabajadores $T = T_1, T_2, \dots, T_n$, donde cada trabajador tiene atributos:
- a_i : Antigüedad del trabajador T_i .
- m_i : Modalidad preferida del trabajador T_i (1 para presencial, 2 para a distancia).
- c_i : Curso preferido del trabajador T_i .
- Una lista de m cursos $C = C_1, C_2, \dots, C_m$, donde cada curso tiene atributos:
- q_j : Cupo máximo del curso C_j .

. d_j : Demanda observada del curso C_j .

Outputs:

- Una función de asignación $f(T)$, que satisface las restricciones:
- $f(T_i) = C_j$ si T_i es asignado al curso C_j .
- $f(T_i) = -1$ si T_i no puede ser asignado.

Restricciones:

1. Cupo Máximo:

$$\forall C_j \in C, \quad \sum_{T_i \in C_j} 1 \leq q_j$$

2. Priorización por Antigüedad:

Si $a_i > a_k$, entonces $f(T_i) \neq -1$ antes que $f(T_k) \neq -1$.

Características de los Datos

Variables Clave

1. Trabajadores:

- $Antigüedad(a_i)$: Representa el tiempo que un trabajador lleva en el servicio profesional. En la base original, algunos valores eran negativos o nulos, lo que requirió limpieza.
- $Modalidad(m_i)$: Puede ser "presencial" o "a distancia". En los datos originales, había errores como nombres mal escritos o inconsistencias entre registros.
- $CursoPreferido(c_i)$: Indica el curso que el trabajador solicitó. Se observó duplicidad de valores en algunos registros.

2. Cursos:

- $CupoMximo(q_j)$: Número máximo de participantes permitidos por curso.
- $Demanda(d_j)$: Número de trabajadores que solicitaron el curso.

3. Dimensiones

- La base de datos incluye aproximadamente n trabajadores y m cursos.
-

Propuesta de Solución

Descripción del Algoritmo Propuesto

Para resolver el problema de asignar trabajadores a cursos respetando las restricciones (cupos, antigüedad, modalidad), usamos un enfoque **combinatorio** basado en listas prioritarias y asignación secuencial. Este enfoque es eficiente y asegura que los trabajadores con mayor antigüedad sean priorizados mientras que los cupos no se excedan.

El algoritmo se divide en tres etapas principales:

1. Preprocesamiento de Datos:

- Limpieza de datos para eliminar registros incompletos o inconsistentes.
- Normalización de valores como modalidades y antigüedad.
- Conversión de las modalidades y cursos en variables categóricas para fácil manejo.

2. Algoritmo

El algoritmo comienza ordenando dos listas importantes:

1. Lista de Trabajadores:

- Se organiza en orden descendente de antigüedad. Esto asegura que los trabajadores con mayor antigüedad sean procesados primero, alineándose con la prioridad establecida en el proyecto.
- Este paso utiliza una función de comparación que evalúa directamente los valores de antigüedad.

2. Lista de Cursos:

- Los cursos están preordenados según su demanda (). Esto garantiza que los cursos más solicitados tengan prioridad implícita al momento de la asignación, evitando que queden recursos subutilizados.

3. Asignación a Grupos

1. Evaluación Secuencial:

- Se recorren los grupos existentes para verificar si es posible asignar al trabajador actual.

2. Condiciones de Asignación:

- El grupo tiene espacio disponible.
- El curso solicitado por el trabajador coincide con el curso asignado al grupo, o bien, el grupo aún no tiene un curso definido.

3. Asignación Exitosa:

- Si se encuentra un grupo adecuado, el trabajador es asignado y se registra su curso en el grupo (si el grupo no tenía un curso asignado previamente).

4. Grupos Nuevos o No Asignados:

- Si no se encuentra un grupo adecuado para el trabajador (por falta de espacio o incompatibilidad de cursos), se registra con un identificador especial , indicando que no pudo ser asignado.

4. Asignación de Solicitudes No Procesadas

Cuando un trabajador no puede ser asignado a un grupo, el algoritmo toma las siguientes acciones:

- Se registra en la lista de no asignados con su información completa: antigüedad, curso solicitado y modalidad.
- Estos datos pueden ser utilizados para futuras iteraciones del programa o para análisis posteriores.

5. Salida de Resultados

El algoritmo produce un diccionario estructurado con los siguientes campos:

- Identificadores de Trabajadores: Incluye el ID único de cada trabajador.
- Estados: Lugar de origen o trabajo del empleado.
- Antigüedad: Tiempo de servicio del trabajador.
- Cursos Solicitados: La preferencia de capacitación del trabajador.
- Identificadores de Grupos: ID del grupo asignado o si no fue posible realizar la asignación.

Pseudocódigo:

Input: Lista de trabajadores T, Lista de cursos C
Output: Asignaciones A, Lista de no asignados NA

1. Limpiar y normalizar los datos de T y C
2. Ordenar T por antigüedad de mayor a menor
3. Inicializar A como lista vacía
4. Inicializar NA como lista vacía
5. Para cada trabajador t en T:


```

Si t.curso_preferido tiene cupo:
    Asignar t a t.curso_preferido
    Reducir cupo en t.curso_preferido
Si no:
    Buscar otro curso c en C compatible con t.modalidad:
        Si c tiene cupo:
            Asignar t a c
            Reducir cupo en c
            Continuar
    Agregar t a NA (no asignado)

```

6. Retornar A y NA

Estrategias de Diseño

1. Divide y Vencerás:

- Separar el problema en pasos manejables: limpieza, ordenamiento y asignación.
- La limpieza de datos se realiza antes de la asignación para evitar errores durante el procesamiento.

2. Prioridad y Greedy:

- El algoritmo sigue una estrategia greedy asignando a los trabajadores por antigüedad de manera secuencial.
- Esta estrategia es eficiente ya que no requiere optimización global.

3. Manejo de Excepciones:

- Si no hay cupo para un trabajador, este se registra como no asignado, manteniendo un registro para futuros ajustes.

Preprocesamiento de Datos

Los datos originales de la encuesta del INE presentaban varios problemas que requirieron un preprocesamiento.

1. Limpieza de Datos:

- **Valores faltantes:** Se eliminaron o imputaron valores nulos en las columnas clave como antigüedad, modalidad, y curso.
- **Duplicados:** Se identificaron y eliminaron registros duplicados.
- **Normalización de campos:** Las modalidades, estados y nombres de los cursos fueron estandarizados para evitar inconsistencias.

2. Transformación de Datos:

- **Categorización:** Las modalidades se codificaron en variables numéricas para facilitar su manejo.
 - **Rango de antigüedad:** Se ajustaron los valores negativos y se agruparon en rangos (Por ejemplo: 0-2 años, 3-5 años).
 - **Demanda por curso:** Se calculó la demanda de cada curso sumando los registros que lo solicitaban, identificando aquellos con alta sobrecarga o baja inscripción.
3. **Preparación para el Algoritmo:**
- Se creó una lista ordenada de trabajadores basada en su antigüedad.
 - Los cursos se organizaron por modalidad y se les asignaron cupos iniciales según la demanda observada.
-

Análisis de Correctitud y Complejidad

Correctitud del Algoritmo

El algoritmo asegura que cada trabajador sea procesado siguiendo las reglas planteadas:

1. Ordenación Inicial:
 - Los trabajadores son ordenados por antigüedad en orden descendente, asegurando que los más antiguos sean procesados primero.
 - Los cursos están preordenados por demanda, priorizando los más solicitados.
2. Condiciones de Asignación:
 - El algoritmo verifica que el grupo tenga espacio disponible y que el curso sea compatible, lo cual respeta las restricciones planteadas.
3. Asignación de No Procesados:
 - Los trabajadores no asignados se registran adecuadamente con identificadores -1.

Complejidad Temporal

1. Ordenación Inicial:
 - Ordenar n trabajadores por antigüedad tiene una complejidad de $O(n \log n)$.
 - La lista de cursos preordenada no contribuye adicionalmente a la complejidad.
2. Asignación a Grupos:

- Para cada trabajador, el algoritmo verifica los grupos disponibles. Si hay g grupos y n trabajadores, este paso tiene una complejidad de $O(n \cdot g)$.

En el caso promedio, g es mucho menor que n , por lo que la complejidad final es aproximadamente $O(n \log n + n \cdot g)$.

Complejidad Espacial

1. Listas y Diccionarios:

- La lista de trabajadores y cursos requiere espacio proporcional a sus tamaños ($O(n + g)$).
- El diccionario de resultados crece linealmente con el número de trabajadores ($O(n)$).

2. Memoria Total:

- La memoria utilizada es $O(n + g)$, ya que no se utilizan estructuras adicionales complejas.
-

Resultados Obtenidos

Distribución de Asignaciones

- **Grupo Formado:** Se formaron **160 grupos** con capacidad máxima de **20 personas** por grupo.
- **Tasa de Asignación:** El algoritmo asignó al **96.7% de los trabajadores** a un grupo, logrando cubrir la mayoría de las solicitudes.
- **Reasignaciones:** Un **2.6% de los trabajadores** fueron reasignados a cursos alternativos dentro de su modalidad.
- **No Asignados: 114 trabajadores (3.3%)** no pudieron ser asignados debido a falta de cupo o incompatibilidad de cursos solicitados.

Saturación de Cursos

- **Cursos Completamente Saturados:** La mayoría de los grupos alcanzaron su capacidad máxima de **20 personas**.
- **Cursos con Espacios Sobrantes:** Algunos grupos quedaron **parcialmente llenos**, como los grupos con **4, 11, 12, y 13** participantes. Estos representan oportunidades para reconfigurar cupos o ajustar la demanda en futuras iteraciones.

Datos de Trabajadores No Asignados

- **Número Total: 114 trabajadores** no fueron asignados a ningún grupo.
- **Principales Razones:**
 - **Falta de espacio** en los cursos solicitados.
 - **Incompatibilidad** entre los cursos solicitados y las modalidades disponibles.
- Estos trabajadores han sido registrados en una lista para **análisis futuro**, priorizando su asignación en próximas iteraciones del programa.

Eficiencia del Algoritmo

- **Configuración Utilizada: 160 grupos** con un máximo de **20 personas** por grupo.
- **Tiempo de Procesamiento:** El algoritmo completó la asignación de aproximadamente **3,400 trabajadores** en menos de **1 minuto**.
- La eficiencia del algoritmo permitió manejar grandes volúmenes de datos, manteniendo una asignación precisa y respetando las restricciones definidas.

Vista Detallada de un Grupo Específico

	id_trabajador	estado	antigüedad	curso	curso_id
23	1258	México	32.0	Selección de personal	9
117	385	Jalisco	31.0	Selección de personal	9
118	754	Ciudad de México	31.0	Selección de personal	9
119	801	Guanajuato	31.0	Selección de personal	9
120	1030	México	31.0	Selección de personal	9
121	1200	Aguascalientes	31.0	Selección de personal	9
122	591	Guerrero	31.0	Selección de personal	9
123	1067	México	31.0	Selección de personal	9
124	1190	Guanajuato	31.0	Selección de personal	9
125	1342	Yucatán	31.0	Selección de personal	9
347	302	Guanajuato	30.0	Selección de personal	9
348	596	México	30.0	Selección de personal	9
349	791	Campeche	30.0	Selección de personal	9
350	1598	Jalisco	30.0	Selección de personal	9
351	1732	Nuevo León	30.0	Selección de personal	9
352	2099	Zacatecas	30.0	Selección de personal	9
353	6	Chihuahua	30.0	Selección de personal	9

El archivo **asignaciones.csv** fue generado y contiene toda la información sobre los trabajadores, los cursos y los grupos asignados. Esto incluye detalles sobre los trabajadores no asignados.

Interpretación de los Resultados

Los resultados obtenidos reflejan que el algoritmo cumplió con los objetivos planteados:

- **Objetivo de Asignación Eficiente:** Se logró asignar al 96.7% de los trabajadores, lo que indica que el algoritmo es eficaz para procesar grandes volúmenes de datos y respetar las restricciones de cupo y modalidad.
- **Priorización por Antigüedad:** Los trabajadores con mayor antigüedad fueron asignados con éxito a sus cursos preferidos, mostrando que las reglas de priorización fueron implementadas correctamente.
- **Manejo de Trabajadores No Asignados:** Aunque un 3.3% de los trabajadores quedaron sin asignar, este grupo fue identificado y registrado, lo que permite considerarlos en iteraciones futuras o ajustar los cupos en cursos demandados.

Limitaciones del Enfoque

A pesar de su efectividad, el enfoque presenta algunas limitaciones:

- **Grupos Parcialmente Llenos:** Algunos cursos tuvieron grupos con cupos sobrantes, lo que podría indicar una distribución no óptima de recursos.
- **Incompatibilidad de Modalidades:** Trabajadores que solicitaron modalidades específicas (e.g., "a distancia") quedaron sin asignar debido a la falta de oferta en esos cursos.
- **Datos Dependientes de Limpieza:** La calidad de los resultados depende directamente de la limpieza inicial de los datos. La base original requería una gran cantidad de correcciones para ser utilizable, lo que podría no ser viable en tiempo real.
- **Demanda Desbalanceada:** Algunos cursos tienen demanda excesiva mientras que otros quedan subutilizados. Esto podría ajustarse con un análisis más profundo de las tendencias de preferencia.

Conclusiones y Trabajo Futuro

Conclusiones Principales

- El algoritmo cumplió con el objetivo de asignar trabajadores a cursos de manera eficiente, respetando restricciones clave como cupos y priorización por antigüedad.
- Se generó un archivo estructurado que permite un análisis posterior de asignaciones, no asignados y saturación de cursos.
- La implementación demostró ser escalable, procesando más de 3,400 registros en menos de un minuto, con resultados consistentes y alineados con las metas del proyecto.

Posibles Extensiones

1. Mejora en la Distribución de Grupos

- **Optimización:** Implementar técnicas como programación lineal para minimizar los cupos sobrantes en los grupos.

2. Ajustes a la Oferta de Cursos

- **Análisis de Patrones:** Examinar la demanda para ajustar los cupos y modalidades de los cursos, reduciendo así las incompatibilidades.

3. Automatización de la Limpieza de Datos

- **Herramientas Automatizadas:** Desarrollar herramientas que automaticen la normalización y limpieza de datos, disminuyendo la dependencia de procesos manuales.

4. Evaluación de Desempeño

- **Métricas Adicionales:** Incluir nuevos indicadores para evaluar la satisfacción de los trabajadores asignados y el impacto real de los cursos completados.

5. Asignación Predictiva

- **Machine Learning:** Utilizar técnicas de aprendizaje automático para predecir la demanda de cursos y ajustar los cupos antes de iniciar el proceso de asignación.
-

Bibliografía

- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to Algorithms. MIT Press.
- INE (2023). Encuesta de Necesidades de Capacitación. Dirección Ejecutiva del Servicio Profesional Electoral Nacional.
- McKinney, W. (2010). Data Structures for Statistical Computing in Python. Proceedings of the 9th Python in Science Conference.
- Pandas Documentation. (2024). pandas.pydata.org. <https://pandas.pydata.org/>
- Python Software Foundation. (2024). Python 3.10 Documentation. <https://docs.python.org/3.10/>