















Última modificación oficial: 26 feb 2026

Para una mejor experiencia usamos la suite de Google.

Técnicas usadas: Outline + Bulletpoints + Estilo Cornell + Tablas y gráficos visuales

Puedes borrar el contenido de la sección de palabras clave o anotaciones para que puedas escribir las tuyas. ¡Éxitos!

Índice

S3: Modelos de Producción (2 Índices) y Formulación Compacta	2
① ¿POR QUÉ UNA SOLA LETRA YA NO ES SUFICIENTE PARA TOMAR DECISIONES?	2
 Lógica del concepto:	2
 Mecanismo / Proceso:	2
 Reglas / Condiciones:	3
 Resumen Express:	3
② ¿CÓMO CONDENSAMOS MIL RESTRICCIONES EN UNA SOLA LÍNEA DE ECUACIÓN?	3
 Lógica del concepto:	3
 Mecanismo / Proceso:	4
 Reglas / Condiciones:	4
 Resumen Express:	5
③ ¿CÓMO GARANTIZAMOS QUE LA MÁQUINA NO INVENTE NÚMEROS MÁGICOS?	5
 Lógica del concepto:	5
 Mecanismo / Proceso:	5
 Reglas / Condiciones:	5
 Resumen Express:	6



S3: Modelos de Producción (2 Índices) y Formulación Compacta

FICHA TÉCNICA:

- Curso:** Investigación de Operaciones I
- Unidad:** Modelos de Producción y Notación Compacta
- Objetivo:** Dominar el modelamiento matricial saltando de variables unidimensionales a bidimensionales (X_{ij}) sin errores de sintaxis lógica.

[PROGRESO]:  19% Completado (Semana 3 de 16)

Las calculadoras mueren cuando enfrentan la vida real. En Wall Street o en las plantas automotrices de General Motors, los ingenieros no toman decisiones "producto por producto"; diseñan redes matriciales masivas que procesan miles de variables simultáneamente. Hoy dejamos atrás el plano 2D y entramos a las grandes ligas de los datos.

Al terminar esta sesión, serás capaz de traducir problemas logísticos y de producción más complejos (con múltiples productos, máquinas y destinos) en unas cuantas líneas de código matemático elegante, ahorrando horas de escritura en exámenes y dejándote listo para programar en Python/Gurobi.

1 ¿POR QUÉ UNA SOLA LETRA YA NO ES SUFICIENTE PARA TOMAR DECISIONES?

El salto a los 2 Índices (Matriz de Decisión)

Lógica del concepto:

Cuando una decisión de negocios depende de dos factores que interactúan al mismo tiempo (ej. *Qué fabrico y dónde lo fabrico*), la variable tradicional (X_i) se rompe por falta de especificidad. Necesitamos crear una matriz. Aquí nace la variable bidimensional X_{ij} , donde cada subíndice representa una dimensión distinta e independiente del problema.

Mecanismo / Proceso:

1. **Identifica a los actores:** Ayuda bastante preguntarse de qué depende la acción. (Ej: Origen y Destino / Tipo de leche y Tecnología).
2. **Bautiza tus índices (Declaración):** A eso le asignaremos una letra i a la primera dimensión y establece su rango (ej: $i = 1, 2$ para tipo de producto) y j a la segunda dimensión (ej: $j = 1, 2, 3$ para tipo de

Palabras Clave/ Anotaciones

Qué tal! Espero te sirva este material. Si tienes dudas u observaciones, date una pasada por la página web y contáctame!

Las primeras dos semanas fueron una introducción al curso, a partir de ahora son los temas que estarán presentes durante todo el curso.

Es decir que pasamos de solo un índice a dos. Ahora ya no es solo producto A sino que también más factores como cliente B o local C

En verdad no tiene por qué ser i o j , podrían ser cualquier letra pero por notación y costumbre utiliza esas



máquina).

3. **Estructura la variable central (X_{ij}):** Describe *exactamente* la acción conjugando ambos actores. (Ej: "Cantidad de litros de leche i procesados usando la tecnología j ").

Reglas / Condiciones:

- **Interacción innegociable:** La variable de decisión *debe obligatoriamente* contener ambos subíndices para ser considerada de 2 índices.
- **Escalabilidad algorítmica:** Los datos numéricos del problema (costos, capacidades, tiempos) ahora mutan. Pueden ser vectores unidimensionales (ej: $Costo_i$) o matrices de interacción bidimensional (ej: $TiempoConsumido_{ij}$).

⚠ **La Trampa:** Olvidar la unidad de medida temporal o física en tu declaración de variables. Poner "Cantidad de producto i en máquina j " te restará puntos. Escribir "Kilogramos de producto i procesados en la máquina j por semana" te asegura el puntaje completo. No te arriesgues!

Resumen Express:

Tipo de Variable	¿Cuándo se usa?	Ejemplo visual de la decisión
Simple (X_{ij})	Decisiones de una vía.	¿Cuántos escritorios de tipo i fabrico?
Matricial (X_{ij})	Decisiones combinadas.	¿Cuántos escritorios tipo i fabrico en la planta j ?

Esto vendría ser tu variable de decisión de forma explícita, donde detallas la acción y las unidades de tu letra "i" o "j"

Siempre escribe la declaración completa, aún así se "sobreentiende" creeme que el profesor no lo "sobreentiende"

Introducción a la forma compacta.

¿CÓMO CONDENSAMOS MIL RESTRICCIONES EN UNA SOLA LÍNEA DE ECUACIÓN?

 La Formulación Compacta y el uso del "Para Todo" (\forall)

Lógica del concepto:

Si tenemos 50 productos y 10 fábricas, escribir "Lo que entra a la fábrica 1 debe ser menor a su capacidad... y a la 2... y a la 3..." generaría 500 ecuaciones infinitas. La "Notación Compacta" usa el símbolo de sumatoria (Σ) para agrupar consumos, y el símbolo "Para Todo" (\forall) para crear familias enteras de restricciones de una.



⚙️ Mecanismo / Proceso:

1. **Detecta el cuello de botella (El Límite / RHS):** Hay que mirar qué recurso (o exigencia) nos está condicionando. (Ej: Capacidad máxima que tiene la *fábrica j*).
2. **Arma el paquete (La Sumatoria Σ):** En el lado izquierdo de la inecuación, hay que agrupar todos los elementos que chupan ese recurso, sumando sobre el *índice opuesto* (ej: sumamos todos los productos i).
3. **Dispara la orden (El cierre \forall):** Al final de la línea, indica matemáticamente para quiénes aplica esta regla usando el símbolo \forall .

📏 Reglas / Condiciones:

- El índice que "corre" debajo de la sumatoria Σ **nunca** debe ser el mismo índice que va después del \forall .
- El símbolo \forall va exclusivamente al final de la restricción. Si falta, Gurobi (y tu profe) no sabrán cuántas ecuaciones debían generar.

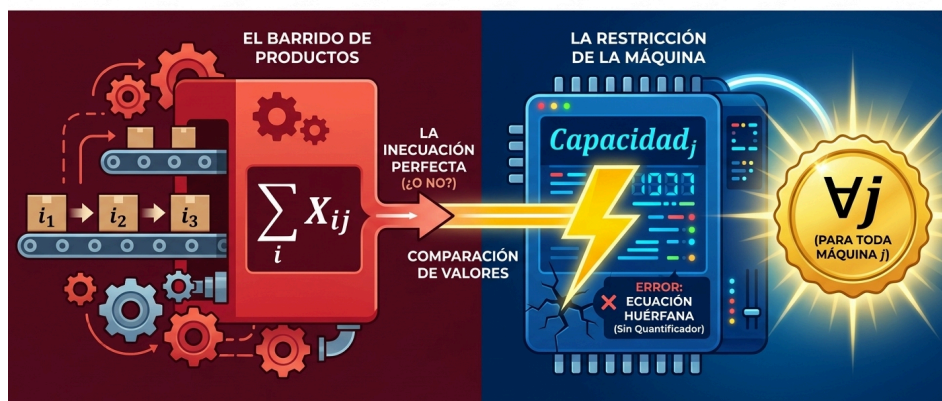
🔒 Mnemotecnia: "La Regla del Espejo RHS (Right Hand Side)"

¿Te confundes al armar la ecuación en el examen? Sigue este truco infalible:

- **Paso 1:** Mira el parámetro de tu Lado Derecho (RHS).
- **Paso 2:** Si el Lado Derecho tiene el índice " j " (ej. *Capacidad_j*)
- **Paso 3:** La sumatoria del lado izquierdo OBLIGATORIAMENTE barre el índice contrario (Σ_i).
- **Paso 4:** El final de la línea OBLIGATORIAMENTE es un espejo del lado derecho: termina en \forall_j .
- ¡Lo que define al límite, define al Para Todo!

⚠️ **La Trampa:** Hacer la inecuación perfecta pero olvidarte de escribir el \forall_j al final. Para la lógica computacional, una ecuación sin "Para todo" es una ecuación huérfana que no significa nada. El modelo entero marcará error.

LA TRAMPA DEL MODELADO: OLVIDAR EL "PARA TODO ($\forall j$)"



En la lógica computacional, una restricción como $\sum_i X_{ij} \leq Capacidad_j$ sin el cuantificador $\forall j$ carece de significado. El modelo no sabe para qué máquina j aplicar la regla y fallará. El sello $\forall j$ valida la ecuación para todas las máquinas j

Dependiendo del caso pero normalmente es: Si se trata de sumatoria " i " entonces aplica para todo " j " y viceversa.

Si pusiste en tus parámetros algo como *Productos_j*, entonces automáticamente debes pensar que eso es "para todo j " al final, por ende la sumatoria debe ser de " i "



💡 Resumen Express:

Lado Izquierdo (Acumulador)	Signo	Lado Derecho (Límite RHS)	Cierre (Familia \forall)	Significado Conductual
$\sum_i X_{ij}$ (Sumamos productos)	\leq	<i>Capacidad_j</i>	$\forall j$	"Para CADA planta j, lo que fabrica no vuela su límite."
$\sum_j X_{ij}$ (Sumamos plantas)	\geq	<i>Demanda_i</i>	$\forall i$	"Para CADA producto i, lo que reunimos cubre al cliente."

3 ¿CÓMO GARANTIZAMOS QUE LA MÁQUINA NO INVENTE NÚMEROS MÁGICOS?

📌 La Matriz FO y las Condiciones de Borde (Signo)

🧠 Lógica del concepto:

Un algoritmo de optimización es ciego. Si no le ponemos límites sanos básicos en la Función Objetivo (FO) y en el cierre del modelo, nos podría sugerir "fabricar cantidades negativas de sillas" porque matemáticamente eso abarataría costos. Debemos enjaular a la fiera con doble sumatoria y leyes de no-negatividad.

⚙️ Mecanismo / Proceso:

1. **La Doble Sumatoria de la FO:** Para maximizar utilidades (o minimizar costos), se debe barrer toda la matriz multiplicando el parámetro económico (Ganancia/Costo) por la variable de decisión.
2. **El candado final:** Agregar la restricción de positividad.

📏 Reglas / Condiciones:

- **Estructura FO:** $MAX Z = \sum_i \sum_j (Utilidad_{ij} \times X_{ij})$. (Se usa doble sumatoria porque hay que sumar filas y columnas de toda la matriz $i \times j$).
- **Condición de No-Negatividad:** OBLIGATORIO escribir siempre al final

Casi siempre cuando se formula la función objetivo de forma compacta se usa doble sumatoria, casi siempre.



del modelo: $X_{ij} \geq 0 \forall i \forall j$.



Resumen Express:

El código pro para un modelo compacto de 2 índices siempre debe tener estas 3 partes visibles para aprobar:

1. **FO:** (Doble $\Sigma\Sigma$)
2. **Sujeto a (Restricciones):** (Una Σ por familia + Cierre \forall de espejo).
3. **Leyes Físicas:** (Declaración de $X_{ij} \geq 0$).

Genial! Si llegaste hasta aquí es más que suficiente para entender la semana. Pero si quieres profundizar alguna parte o concepto puedes visitar los recursos adicionales en www.ingenierolocaso.com en la sección de CONTENIDO.

Utiliza ese material como guía o complemento, de todas formas tienes que darle una repasada, ESE EXÁMEN NO SE VA A PASAR POR ARTE DE MAGIA. TU PUEDES!