Silvana Sandoval Galvis — 202123682

Juan Felipe Camelo — 202122062

**Informe de laboratorio**

**Problema 1**

**Modelo Matemático**

**Ecuación objetivo**

Esta ecuación busca maximizar el número de tareas a realizar con base en la prioridad que estas tienen. En esta, representa todas las tareas disponibles, mientras que puntosPrioridad representa que tan prioritarias son las tareas para realizar. Si una tarea tiene 1 punto prioritario, esto es equivalente una prioridad mínima. Por el contrario, si una tarea tiene 7 puntos de prioridad esto es equivalente a que haya una prioridad máxima.

**Restricciones**

Esta ecuación busca restringir el número de puntos de historias de usuario que pueden realizar los 4 desarrolladores. Estos, en total, tiene 52 puntos de historia disponibles, por lo cual se busca que la suma de los puntos de historia de usuario de las tareas que se asignen no superen este número. En esta, representa todas las tareas disponibles, mientras que puntosHistoria representa los puntos de historias de usuario que cada tarea tiene.

En el anterior modelo se puede apreciar la función objetivo donde se busca maximizar el número de tareas con la mayor prioridad para que los 4 trabajadores lo hagan. Para poder representar esto en el código, este se representó de la siguiente manera:

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

En la imagen se puede apreciar que se decidió que la variable *x* sería una variable binaria ya que permitirá indicar cuales tareas se representaran (con 1) y cuales no (con 0). Además, se modelaron los puntos de historia y los de prioridad como diccionarios ya que esto nos permite asignarle los valores correspondientes a cada tarea. Por último, se decidió escribir tanto la función objetivo como la restricción con la función *sum* ya que es aquella que permite representarlas.

**Resultados**

**Interfaz de usuario gráfica, Texto

Descripción generada automáticamente con confianza media**

Esos resultados muestran las tareas que fueron asignadas con el propósito de maximizar los puntos de prioridad. Para este fin, se asignaron las tareas 1, 2, 3, 4, 6, 7, 8, 9 y 10 que, en total, tendrán 52 puntos de historia, lo cual respeta la restricción de la totalidad de puntos de historia que pueden llevar a cabo todos los desarrolladores. Por otra, parte la prioridad total de estas tareas es de 46 puntos de prioridad, el cual es el máximo que se puede alcanzar, así llegando a la solución óptima.

**Problema 2**

**Modelo matemático y decisiones de diseño**

Para empezar, la función objetivo se modeló de la siguiente manera:

Esta ecuación busca maximizar la ganancia generada por las tareas asignadas a cada trabajador. En esta *ganancia[i]* representa la ganancia que generará cada una de las tareas que pueden ser asignadas. Por otra parte, representa todas las combinaciones de empleados y tareas que se busca asignar.

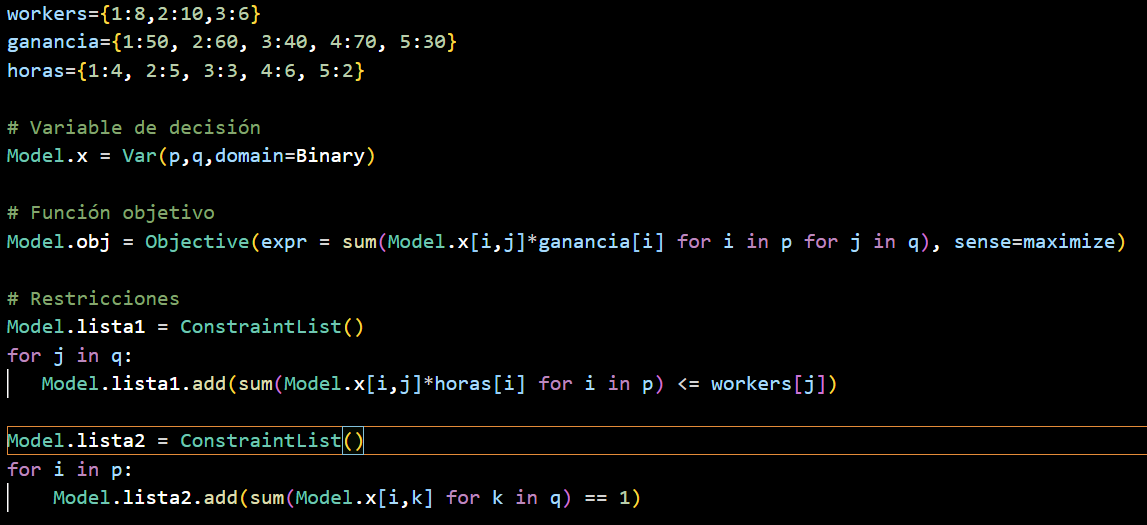
Para este problema se tomó la decisión de separar cada requerimiento en una restricción diferente para el problema. Esto se hizo con el propósito de poder alcanzar los objetivos propuestos en el enunciado del problema. La primera restricción se ve modelada por la siguiente ecuación:

Esta ecuación modela la restricción donde los trabajadores no pueden tener un número de horas de tareas que supere la disponibilidad que estos tienen para completarlas. En esta, representa la variable de decisión, mientras que *horas* representa las horas que toma realizar cada una de las tareas. Por último, workers[i] representa las horas disponibles totales que tiene un trabajador.

Por otra parte, la segunda restricción se ve modelada por la siguiente ecuación:

Esta ecuación modela la restricción donde se especifica que cada tarea puede ser realizada por 1 solo trabajador. Esto se hace con el fin de que a la hora de maximizar, no se le asignen las mismas tareas a diferentes trabajadores con el fin de tener una ganancia máxima.

Por último, la siguiente imagen muestra el código que se uso para poder encontrar el máximo de la función objetivo. En este, se tomo la decisión de meter las diferentes restricciones en *ConstrainList* para asegurar que estas no se sobrescriban cada vez que se ejecuta el código. Esto se hico para asegurar que cada tarea fuera asignada a 1 solo trabajador. Por otro lado, la variable de decisión se modelo como binaria ya que esta indicaría que tarea realizaría cada trabajador con 1, y las que no con 0.

****

**Resultados**

### **Interfaz de usuario gráfica Descripción generada automáticamente con confianza media**

Estos resultados muestran las tareas se asignaron a cada empleado. El empleado 1 tuvo asignada el trabajo 4 que le tomará 6 horas en realizar, lo cual está dentro de su disponibilidad de 8 horas. Además, el empleado 2 tiene asignadas los trabajos 1 y 2 que, en total, le tomarán 9 horas en realizar, lo cual se encuentra dentro de su disponibilidad de 10 horas. Por su parte, al empleado 3 se le asignaron los trabajos 3 y 5 que en total le ocuparan 5 de sus 6 horas disponibles. Estos resultados cumplen las restricciones de tiempo y de asignación de tareas ya que cada tarea se le asignó a 1 solo empleado y el tiempo que les tomará en realizar sus tareas es menor al tiempo disponible. Finalmente, la ganancia que se generó fue de 250 USD, la cual es la máxima ya que abarca la ganancia de la totalidad de trabajos.

**Problema 3**

**Modelo Matemático y decisiones de diseño**

Para este problema se tomó la decisión de separar cada requerimiento en diferentes restricciones para que así fuera más claro el objetivo de cada restricción y fuese fácil de modificar. La primera restricción nos asegura que cada avión solo lleve la cantidad de peso que puede llevar.

Donde A es el conjunto de los aviones, R el conjunto de los recursos, **x** el modelo y **c** es la capacidad (peso) del avión j.

La segunda restricción nos asegura que cada avión solo lleve los recursos que le quepan (volumen).

Donde **cv** es la capacidad volumétrica del avión j.

Ahora, sabemos que el avión 1 no puede transportar medicamentos. La restricción 3 nos asegura que esto se cumpla.

Donde i=2 representa el elemento 2 de R (medicamentos) y j=1 representa el elemento 1 de A (avión 1).

La restricción 4 nos asegura que no haya contaminación cruzada entre equipos médicos y agua potable.

Acá restringimos la suma a los elementos 3 y 4 de R, que son precisamente los equipos medicos y el agua potable. Si ambos elementos se encuentran en el mismo avión, la suma de estos elementos sería mayor a 1, por lo cual pedir que la suma sea 0 o 1 nos asegura que no haya contaminación cruzada.

Finalmente, diseñamos una restricción para asegurarnos de que el mismo recurso no se esté asignando a diferentes aviones.

Lo anterior se puede apreciar plasmado en código de la siguiente manera:

**Texto

Descripción generada automáticamente**

**Resultados:**

* **Caso Base**

Gráfico, Gráfico de barras

Descripción generada automáticamente

Para el caso base se asignaron alimentos básicos al avión 1, ocupando 15 toneladas de 30 toneladas máximas y unos 8 m^3 de los 25 m^3 disponibles. Las medicinas, agua potable y mantas se asignaron al avión 2, que en total ocupan 33 toneladas de las 40 disponibles y unos 20 m^3 de sus 30 m^3 disponibles. Finalmente, los equipos médicos se asignaron al avión 3, que ocupa 20 toneladas de las 40 disponibles y 10 m^3 de los 30 m^3 disponibles.

* **Caso sin restricción de almacenamiento de medicamentos**

Gráfico, Gráfico de barras

Descripción generada automáticamente

Veamos que, sin restricción de almacenamiento de medicamentos, llegamos a la solución que podemos no usar el avión 3. En este caso, se le asigna al avión 1 los equipos médicos y las mantas, que ocupan 30 toneladas de las 30 disponibles y 16 m^3 de los 25 disponibles. En este caso, en el avión 2 transportaríamos alimentos básicos, medicina y agua potable, que ocupan 38 toneladas de las 40 disponibles y 22 m^3 de los 30 disponibles.

* **Caso sin restricción de seguridad del equipo médico**

Gráfico, Gráfico de barras

Descripción generada automáticamente

Sin la restricción, de seguridad del equipo médico veamos que ahora se asigna únicamente los alimentos básicos que ocupan 15 toneladas de las 30 disponibles. Para el avión 2, asignamos los equipos médicos y agua potable, ocupando 38 toneladas de las 40 disponibles y 22 m^3 de los 30 disponibles. Finalmente, se le asigna las medicinas y las mantas que ocupan 15 toneladas de las 35 disponibles y 8 m^3 de los 35 disponibles.

* **Caso sin ninguna restricción de almacenamiento**

Gráfico, Gráfico de barras

Descripción generada automáticamente

Sin restricciones de almacenamiento, se le asigna al avión 1 los equipos médicos y las mantas, que ocupan 16 m^3 de los 25 disponibles. Al avión 2 se le asignan los alimentos básicos, las medicinas y el agua potable, que ocupan 22 m^3 de los 30 disponibles.

* **Caso sin restricción volumétrica**

**Gráfico, Gráfico de barras

Descripción generada automáticamente**