**Aplicación nacional de Programación Entera Mixta**

**Universidad Distrital Francisco José de Caldas**

**LINK DEL REPOSITORIO**

[**https://github.com/IO1-WAO/PEM**](https://github.com/IO1-WAO/PEM)

**Integrantes**

**Andrés Felipe Wilches Torres - 20172020114  
Luis Alejandro Ocampo Gamboa - 20172020050  
Nicolas Andrade Perdomo - 20172020097**

**Presentado a   
  
Alberto Acosta López**

**Tabla de contenidos:**

1. [Resumen](#resumen)
2. [Introducción](#introduccion)
3. [Justificación](#Justificacion_texto)

1. [Objetivos](#Objetivos_texto)

1. [Modelo](#modelo_texto)
2. [Aplicación a Nivel Nacional](#aplicacion_a_nivel_nacional)

6.1) [Ejercicio Aplicado](#aplicacion_ejercicio_aplicado)

6.2) [Manual de Usuario](#aplicacion_manual_de_usuario)

1. [Conclusiones](#conclusiones)
2. [Bibliografía](#bibliografia)

**1)** **Resumen:**

Como hemos visto en el último año el país ha estado en una situación de salud complicada debido a la pandemia COVID-19 la cual es una enfermedad respiratoria causada por un ‎coronavirus recientemente descubierto.

La mayoría de las personas que enferman de ‎COVID-19 experimentan síntomas de leves a ‎moderados y se recuperan sin tratamiento ‎especial [[4]](#Engel_cita) (Engl, N, 2020), pero aquellos que tienen complicaciones médicas, deben ser atendidos en una UCI, por lo tanto, se diseñó una aplicación que utilice las herramientas de la PEM para calcular el costo mínimo de los centros hospitalarios según la cantidad de UCI’s disponibles y su costo de operación.

**2)** **Introducción:**

Desde sus inicios, a mediados del siglo XX, la investigación de Operaciones (OR) ha experimentado un gran aumento en diferentes situaciones de resolución de problemas de múltiples sectores. Uno de estos es el sector salud, en el que se desarrollan los procesos de toma de decisiones principalmente relacionado con la asignación y soluciones de optimización de recursos [[5]](#rome_cita) (Romero-Conrado, 2017).

Con ayuda de esta aplicación los hospitales a nivel nacional van a optimizar el uso de recursos según el tiempo de recuperación de los pacientes, el costo de cada cama y la disponibilidad de estas en cada departamento o centro médico, con el objetivo de minimizar costos.

La capacidad de planificación marca la diferencia entre la vida y la muerte, puede asegurar que la institución tenga la capacidad de responder al nivel de experiencias especialmente en la industria de la salud por que no solo se relacionan con la gestión de recursos altamente especializados y costosos. [[11]](#Vanjauten_cita)

Todas las operaciones relacionadas con el desempeño del servicio de salud en los hospitales están limitadas en términos de capacidad. Por lo tanto, para satisfacer la demanda de atención médica de los pacientes, la dirección del hospital debe tener una planificación y control de la capacidad de las operaciones. [[6]](#Aarabi_cita) [[7]](#Pierskalla_cita) [[9]](#jack_cita)

La aplicación “Manejo de costos para enfermedades”, tiene como finalidad ayudar a las entidades de salud a nivel nacional a calcular el tiempo de recuperación de las personas de acuerdo con la disponibilidad de camas UCI y su capacidad en cada centro hospitalario.[[3]](#zaca_cita) (Zachariah).

El aplicativo funciona con un algoritmo que utiliza el sistema PEM utilizando la información suministrada por cada centro médico o departamento: El número de UCI disponibles, la máxima operabilidad de UCI en cada institución y el costo de operación.

**3)** **Justificación:**

El modelo planteado se basó en un modelo más robusto y complejo de Suriaty Sitepu, nombrado “Optimization Model for Capacity Management and Bed Scheduling for Hospital” [[10]](#sitepu_cita), traducido como: modelo de optimización para el manejo de capacidad y registro de camas para un hospital. En dicho artículo, se hace un breve manejo de los recursos según la disponibilidad de camas, personal disponible y rutas optimas en el viaje entre destinos. El uso de optimización multiobjetivo combinado con simulación para abordar el problema de asignación de UCI’s se puede encontrar en [[12].](#Wang_cita)

Los problemas que ocurren en el mundo real pueden resolverse mediante modelos matemáticos, los problemas pueden modelarse en formas matemáticas que se convierten en ecuaciones [[19]](#Tulus_cita). Los principios matemáticos no solo se utilizan para la sistematización, sino que también funcionan para maximizar los resultados o minimizar los costos [[20]](#Nasution_cita). Los modelos que enfatizan la rentabilidad se centran en la precisión física que se utilizará [[18]](#Alsalemi_cita), y se pueden violar los límites de capacidad dando un valor de penalización [[17].](#Dorling_cita) En el GU et al. se introdujo el modelo del Sistema Ciber físico Médico Fog-Computing-Medical (FC-MCPS), que presenta programación mixta-entera (MLP) sobre cuestiones de costo mínimo con consideración conjunta de asociaciones de usuarios, distribución de tareas y despliegue de VM (máquina virtual) [[16].](#LZeng_cita)  
  
En nuestro país, el costo de una cama UCI ronda los 30.854.200 COP, por un tiempo aproximado de 15 días. [[13]](#Cunningham_cita)

Nuestro modelo parte del planteamiento del artículo de Sitepu [[10]](#sitepu_cita), el cual esta soportado en un modelo parecido pero orientado a tiempos de Pardede [[8]](#pardede_cita). nuestro modelo se adapta a la situación hospitalaria en la ciudad, donde el número de variables dependerá de la cantidad de departamentos o centros hospitalarios para el costo que el usuario desee minimizar.

**4)** **Objetivos:**

El objetivo de nuestra aplicación es crear mediante la programación entera mixta (PEM) [[14]](#AMH_cita) un software que, dada una cantidad de centros hospitalarios en una ciudad, el costo de operabilidad de una cama UCI, su disponibilidad, número máximo de UCI’s por centro se pueda conocer el costo mínimo de operabilidad por centro, basado en un modelo de Sitepu [[10]](#sitepu_cita).

Nuestras variables vienen dadas según la disponibilidad de UCI’s por centro y el costo de estas en el centro, en nuestro caso se manejan las restricciones según el costo por centro y su capacidad.

**5)** **Modelo:**

Explicado lo anterior, la notación es la siguiente:

Conjuntos:

= conjunto de departamentos / centros médicos

Parámetros:

= Fondos disponibles por departamento

= # máxima cantidad de UCI’s por departamento

= # Costo de operación de las UCI’s

Variables de decisión:  
  
 = # UCI’s por departamento / centro médico  
 = # Numero entrante de UCI’s

*Muchas veces no se aumentan el número de UCI’s, por lo que el Puede tener el valor de 0.*  
   
Donde la función objetivo es:  
  
   
Las restricciones son:  
  
 ,  
 ,

Con:

Como vemos, la función objetivo, depende del número de departamentos o centros hospitalarios de los cuales sus recursos se van a minimizar, donde las variables son el número de UCI’s y el número de las mismas entrantes, los parámetros son el costo de operación de dichas UCI’s, los fondos disponibles por departamento / centro médico y la máxima cantidad de UCI’s por instalación.

La primera restricción hace alusión a que el presupuesto gastado no debe ser mayor que el fondo proporcionado mientras que las siguientes vienen dadas por cada hospital, donde el numero disponible de UCI’s no debe superar el máximo que puede alojar.

**6)** **Aplicación a Nivel Nacional:**

Con el modelo explicado anteriormente, se aplicará el modelo a un ejemplo de la ciudad de Bogotá.

**6.1)** **Ejercicio aplicado:**

Se tendrá en cuenta tres hospitales de la ciudad de Bogotá, los cuales son: Unidad de servicios de salud Simón Bolívar, Unidad de servicios de saludo el Tunal y el Hospital San Ignacio.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hospital** | **# UCI’s disp.** | **# UCI’s entrantes** | **Máxima capacidad UCI’s** | **Presupuesto** (COP) |
| Simón Bolívar | 178 | - | 632 | 19.500’000.000 $ |
| Hospital tunal | 75 | - | 668 | 9.480’000.000 $ |
| Hospital San Ignacio | 46 | - | 394 | 5.546’000.000 $ |

**Tabla1. Datos de hospitales seleccionados.** **Fuentes en:** [**[21]**](#InformacionCamas_cita)**[[22]](#SanIgnacio_cita)** **[[23]](#Tunal_cita)** **[[24]](#SimonB_cita)**

Con estos datos obtenidos, se procede a diligenciarlos en el modelo.  
Donde la función objetivo es:  
  
 = 30.854.200 COP

Al reemplazar, la *función objetivo*:

Por lo que queda:

Las restricciones:

Como los fondos disponibles por departamento son constantes, nuestro problema queda de la siguiente forma:

***Función Objetivo:***

***Sujeto a:***

Al reemplazar el valor de bt = 30.854.200 COP, como es un costo se toma negativo, el problema a resolver, finalmente queda de la siguiente forma:

***Función Objetivo:***

***Sujeto a:***

Donde recordemos que las variables son: .

**Solución:**

Al agregar las variables de holgura, nuestro modelo queda de la siguiente forma:

Sujeto a:

Por lo que se procede a solucionarlo en la tabla Simplex:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Iteracion 1 |  | Cj | -54920447600 | -2314065000 | -1419293200 | 0 | 0 | 0 | 0 |  |
| **B** | **Cb** | **Xb** | **XB1** | **XB2** | **XB3** | **S1** | **S2** | **S3** | **S4** | **Xb / XB1** |
| S1 | 0 | 34526000000 | -54920447600 | -2314065000 | -1419293200 | 1 | 0 | 0 | 0 |  |
| S2 | 0 | 632 | (178) | 0 | 0 | 0 | 1 | 0 | 0 | 632 / 178 = 3.5506 |
| S3 | 0 | 668 | 0 | 75 | 0 | 0 | 0 | 1 | 0 |  |
| S4 | 0 | 394 | 0 | 0 | 46 | 0 | 0 | 0 | 1 |  |
| **Z = 0** |  | **Zj** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |  |
|  |  | Zj - Cj | 54920447600 | 2314065000 | 1419293200 | 0 | 0 | 0 | 0 |  |

**Tabla 2. Primera iteración del ejercicio propuesto**

El máximo positivo del Zj – Cj es 5492047600, por lo que la variable entrante es XB1  
El elemento pivote es 178  
Al dividir, el elemento pivote entre el Cj, nos da el valor mínimo, por l que sale S2  
R2 = R2 / 178, R1 = R1 + 5492047600R2, R3 = R3, R4 = R4

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Iteración 2 |  | Cj | -54920447600 | -2314065000 | -1419293200 | 0 | 0 | 0 | 0 |  |
| **B** | **Cb** | **Xb** | **XB1** | **XB2** | **XB3** | **S1** | **S2** | **S3** | **S4** | **Xb / XB2** |
| S1 | 0 | 34526000000 | -54920447600 | -2314065000 | -1419293200 | 1 | 30854200 | 0 | 0 |  |
| x1 | -54920447600 | 3,5506 | 1 | 0 | 0 | 0 | 0,0056 | 0 | 0 |  |
| **S3** | 0 | 668 | 0 | **(75)** | 0 | 0 | 0 | 1 | 0 | 668 / 75 = 8.9067 |
| S4 | 0 | 394 | 0 | 0 | 46 | 0 | 0 | 0 | 1 |  |
| **Z = -19499854400** |  | **Zj** | **-54920447600** | **0** | **0** | **0** | **-30854200** | **0** | **0** |  |
|  |  | Zj - Cj | 0 | 2314065000 | 1419293200 | 0 | -30854200 | 0 | 0 |  |

**Tabla 2. Segunda iteración del ejercicio propuesto**

El mínimo positivo Zj-Cj es 2314065000 y su índice de columna es 2, por lo tanto, la variable que entra es x2

Al dividir el elemento pivote por Zj nos da 8.9067 y su índice de fila es 3, por lo tanto, la variable que sale es S3

El elemento pivote es 75

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Iteración 3 |  | Cj | -54920447600 | -2314065000 | -1419293200 | 0 | 0 | 0 | 0 |  |
| **B** | **Cb** | **Xb** | **XB1** | **XB2** | **XB3** | **S1** | **S2** | **S3** | **S4** | **Xb / XB3** |
| S1 | 0 | 74636460000 | 0 | 0 | -1419293200 | 1 | 30854200 | 30854200 | 0 |  |
| x1 | -54920447600 | 3,5506 | 1 | 0 | 0 | 0 | 0,0056 | 0 | 0 |  |
| x2 | -2314065000 | 8,9067 | 0 | 1 | 0 | 0 | 0 | 0,0133 | 0 |  |
| **S4** | 0 | 394 | 0 | 0 | **(46)** | 0 | 0 | 0 | 1 | 394 / 46 = 8.5652 |
| **Z = -40110460000** |  | **Zj** | **-54920447600** | **-2314065000** | **0** | **0** | **-30854200** | **-30854200** | **0** |  |
|  |  | Zj - Cj | 0 |  | 1419293200 | 0 | -30854200 | -30854200 | 0 |  |

**Tabla 3. Tercera iteración del ejercicio propuesto**

El mínimo positivo Zj-Cj es 1419293200 y su índice de columna es 3, por lo tanto, la variable que entra es x3

Al dividir el elemento pivote por Zj nos da 8.5652 y su índice de fila es 4, por lo tanto, la variable que sale es S4

El elemento pivote es 46

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Iteración 4 |  | Cj | -54920447600 | -2314065000 | -1419293200 | 0 | 0 | 0 | 0 |  |
| **B** | **Cb** | **Xb** | **XB1** | **XB2** | **XB3** | **S1** | **S2** | **S3** | **S4** |  |
| S1 | 0 | 86793014800 | 0 | 0 | 0 | 1 | 30854200 | 30854200 | 30854200 |  |
| x1 | -54920447600 | 3,5506 | 1 | 0 | 0 | 0 | 0,0056 | 0 | 0 |  |
| x2 | -2314065000 | 8,9067 | 0 | 1 | 0 | 0 | 0 | 0,0133 | 0 |  |
| x3 | -1419293200 | 8,5652 | 0 | 0 | 1 | 0 | 0 | 0 | 0,0217 |  |
| **Z = -52267014800** |  | **Zj** | **-54920447600** | **-2314065000** | **-1419293200** | **0** | **-30854200** | **-30854200** | **-30854200** |  |
|  |  | Zj - Cj | 0 |  | 0 | 0 | -30854200 | -30854200 | -30854200 |  |

**Tabla 4. Tabla final del ejercicio propuesto**

Como resultado final, nos queda lo siguiente:

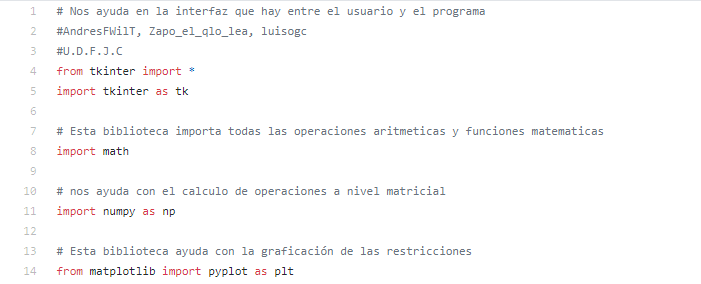
**Interpretación:**Como se apreció, el costo mínimo se alarga los -52.267’014.800, mientras que la cantidad de camas por hospital con las que se obtiene un costo de operación mínimo son:  
 Hospital Simón Bolívar: 4, Hospital Tunal = 9, Hospital San Ignacio = 9

**6.2)** **Manual de Usuario**

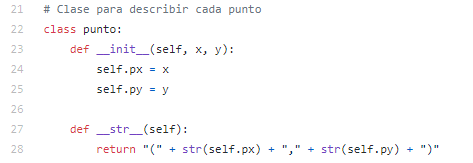
En las primeras líneas de código se importan las librerías que vamos a utilizar en el programa, la primera es tkinter, la cual referenciamos en una variable “tk” para poder utilizar las funciones de dicha librería.

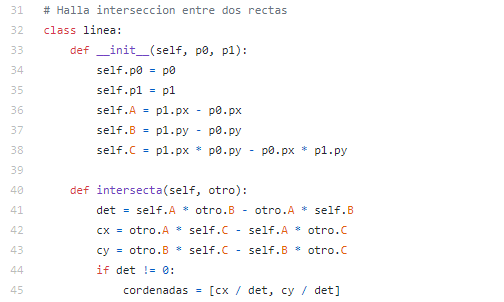
La segunda librería que importamos es math, la cual se utiliza más adelante para realizar unos cuantos cálculos matemáticos.

Por último, importamos la librería numpy referenciada en la variable “np”, la cual usamos para otros cálculos matemáticos más específicos necesarios para la solución de cualquier problema.

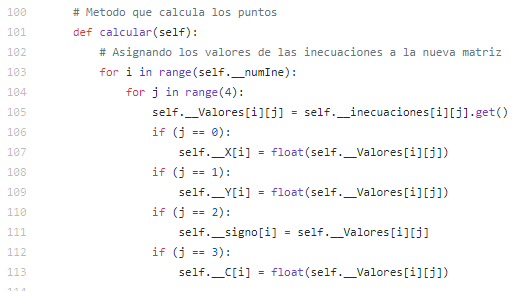
  
**imagen 1, librerías importadas**

Seguido de importar las librerías, creamos una clase llamada “punto” en la cual le definimos su constructor y un método que nos retorna los valores que vaya a ingresar el usuario.

  
**imagen 2, coordenadas de la recta**

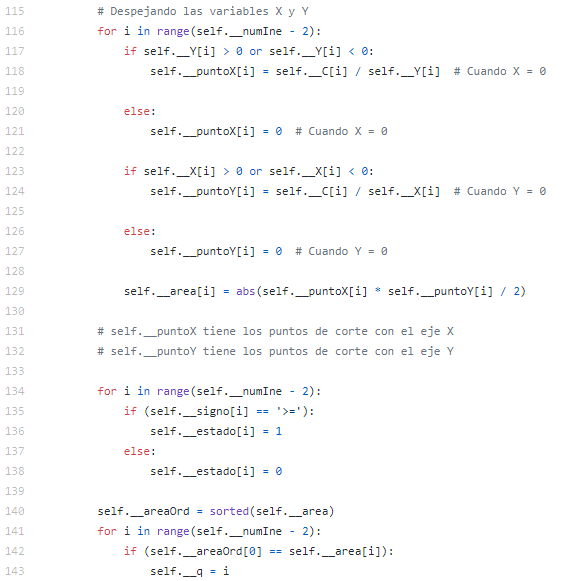
  
**imagen 3, hallar la intersección**

En otra clase distinta, creamos el método “calcular” que sirve para asignar los valores de las inecuaciones ingresadas por el usuario a una matriz.

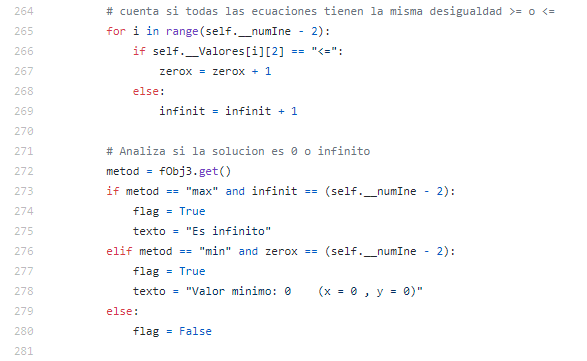


**imagen 4, Método que calcula los puntos**

En el mismo método, hay un bloque de código que sirve para despejar las variables de las inecuaciones.

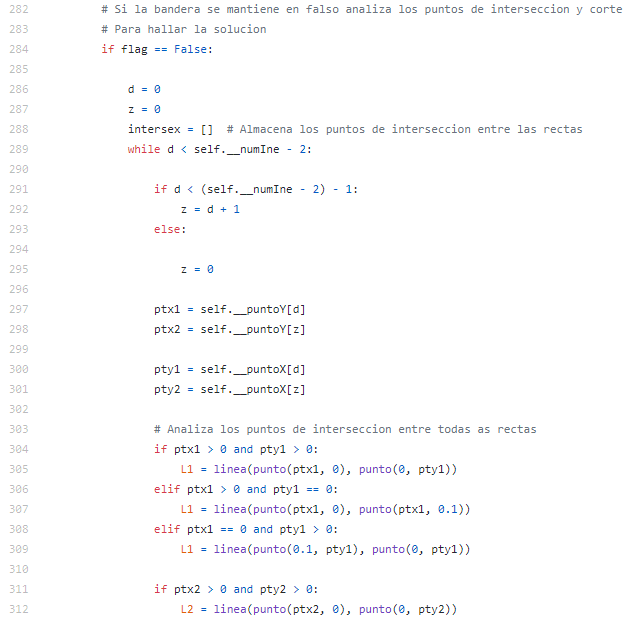
  
**imagen 5, despejar variables**

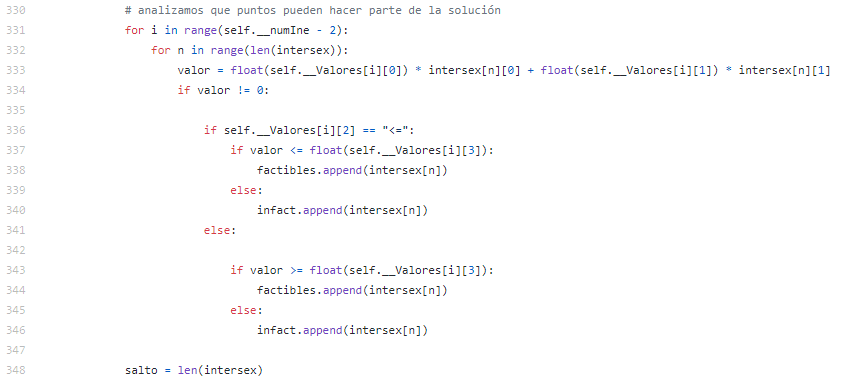
En el siguiente bloque de código contabilizamos los distintos tipos de desigualdades en las inecuaciones, las que sean “<=”, o “>=”, además, analiza si la solución es 0 o infinito.



**imagen 6, analiza la solución**

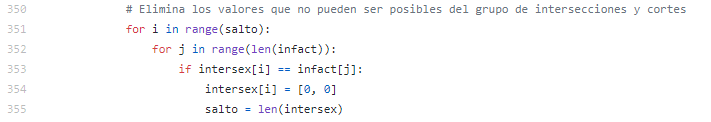
Utilizamos una variable llamada “flag” de tipo booleana para saber que soluciones que se calcularon anteriormente se deben analizar.

  
**imagen 7, analiza los puntos de intersección**

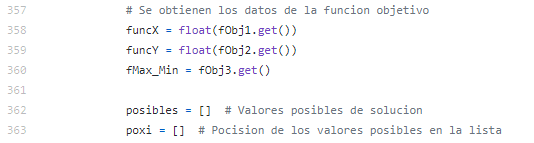


**imagen 8, analiza los puntos que hacen parte de la solución**

Una vez analizados los valores que vamos a necesitar para la solución, se eliminan de la matriz aquellos que no sirven.

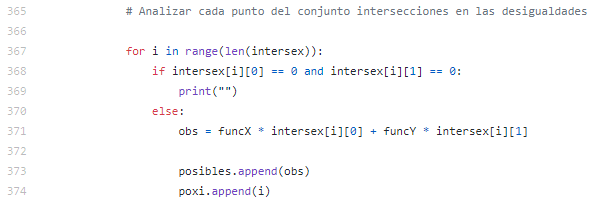
  
**imagen 9, elimina valores que no pueden ser parte de la solución**

En este bloque de código se almacenan los valores que ingreso el usuario en la interfaz gráfica, más específicamente en la función objetivo

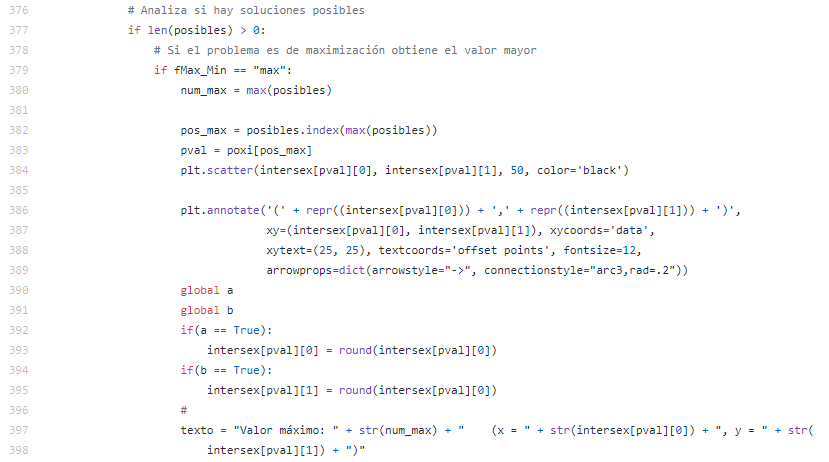


**imagen 10, datos de la función objetivo**

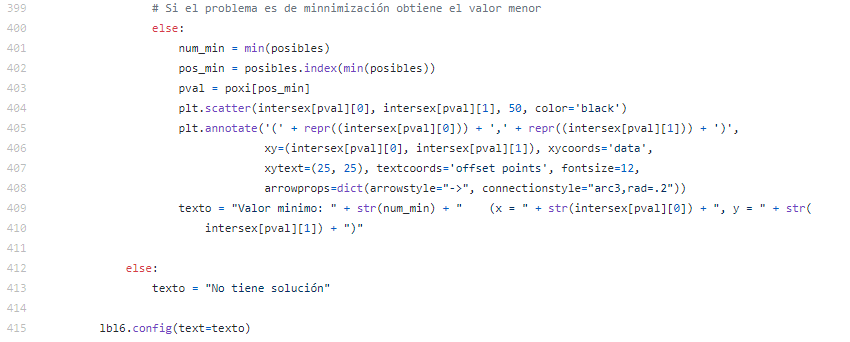
En este punto analizamos el conjunto de valores restantes y los agregamos en otra matriz con los posibles valores solución.

  
**imagen 11, analiza el conjunto de intersecciones**

De los valores que recopilamos en la matriz anterior, se analiza cada uno de ellos para saber si es solución posible para maximización.

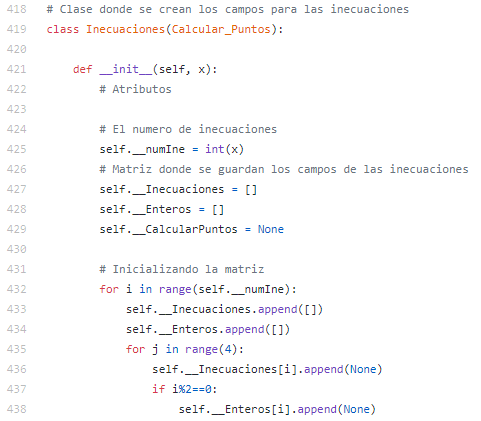
  
**imagen 12, valida la optimización del problema**

En caso de que el problema no sea de maximización, hay dos “else“ en el método para obtener el valor menor, que corresponde a la solución para minimización, o, para indicar que el problema no tiene solución.



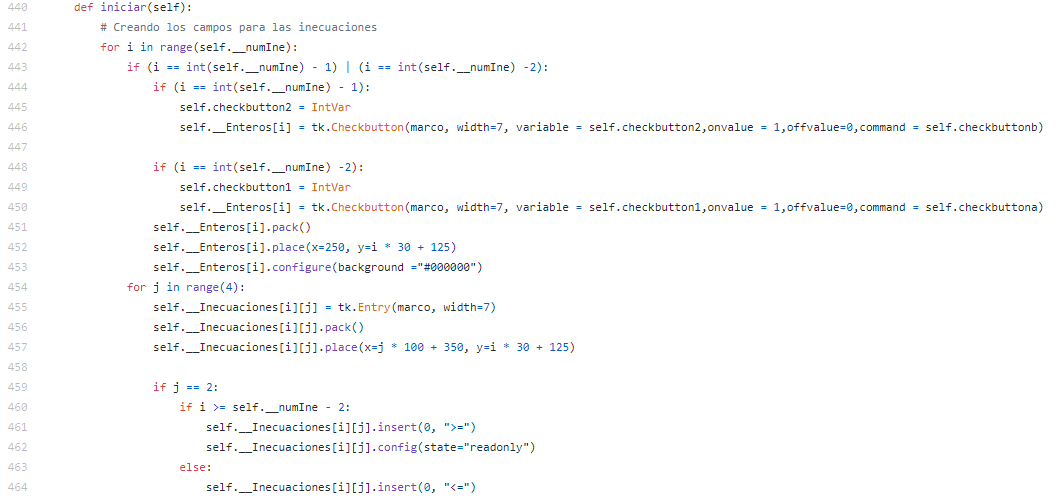
**imagen 13, valida la optimización del problema**

En esta clase llamada inecuaciones almacenamos todos los valores ingresados por el usuario y se guardan en matrices para utilizar en los métodos descritos previamente.



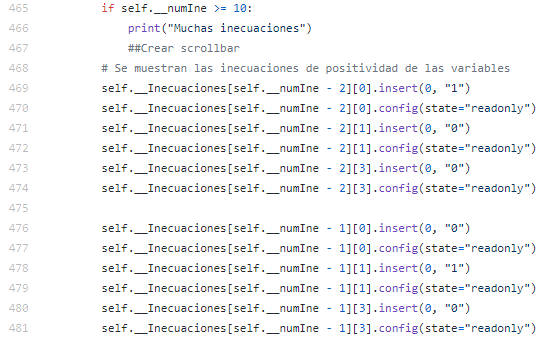
**imagen 14, crea campos de inecuaciones**

En este método “iniciar” se define parte de la interfaz grafica donde el usuario va a ingresar los valores correspondientes a las restricciones.



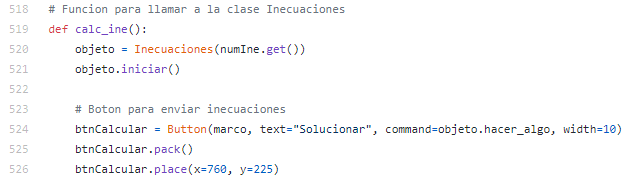
**imagen 15, inserta campos de inecuaciones**

Además de la interfaz donde el usuario ingresa los valores necesarios para el problema, se definen las inecuaciones de no negatividad en el conjunto de restricciones.



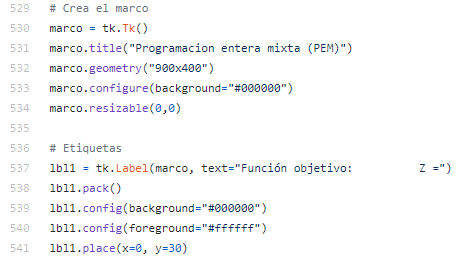
**imagen 16, inserta campos de inecuaciones de positividad**

En este método se invoca la clase “Inecuaciones” y se le envían los valores obtenidos en la interfaz de usuario.



**imagen 17, envía las inecuaciones**

Y el resto del código consta de detalles estrictamente gráficos y que no afectan la lógica del programa, como por ejemplo la creación del marco, las etiquetas y botones, entre otros.

  
**imagen 18, crea el marco**

**Uso de la aplicación**

El programa arrojara la respuesta en una ventana indicando la solución del z y sus respectivas variables.

**7)** **Conclusiones:**

La programación entera mixta se puede solucionar por distintos métodos, por ejemplo, si tenemos dos variables es posible solucionarlo por el método gráfico, y si disponemos de más variables se puede por método simplex simple, simplex dual, la gran M, ramificación y acotamiento, etc.

Los datos al ser comparados con el solucionador de AtozMath [[1]](#atoz_biblio), que arrojaron los mismos valores, por lo que el programa acierta en su solución.

Como resultado de este documento y aplicación, se podría desarrollar un aplicativo que incorpore mas restricciones, por ejemplo, según el numero de personal disponible o cuartos en hospitales [[8]](#pardede_cita).

**8)** **Bibliografía:**

* [[1]](#atoz_texto) *Atozmath CBOM (Simplex)*. (2005). Software. Recuperado de <https://cbom.atozmath.com>

* [[2]](#taha_texto) Taha, H. A. (2000). *Investigación De Operaciones* (5.a ed., Vol. 1). Fayeteville, Estados Unidos: Pearson Educación.

* [[3]](#zaca_tx) Zachariah, R. (s. f.). MDPI. Recuperado 5 de febrero de 2021, de <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjf45SQ_dPuAhXKo1kKHVQHCm0QFjABegQIAhAC&url=https%3A%2F%2Fwww.mdpi.com%2F2414-6366%2F5%2F3%2F118%2Fpdf&usg=AOvVaw0fpxz51C_E06WSbR2U3VJv>

* [[4]](#Engel_tx) Engl, N., & Med, J. (2020, 8 octubre). Remdesivir for the Treatment of Covid-19 — Final Report. PubMed Central (PMC). <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7262788/>

* [[5]](#rome_tx) [5] Romero-Conrado, A. R., Castro-Bolaño, L. J., Montoya-Torres, J. R., & Jiménez Barros, M. Á. (2017). Operations research as a decision-making tool in the health sector: A state of the art. DYNA, 84(201), 129. https://doi.org/10.15446/dyna.v84n201.57504

* [[6]](#Aarabi_texto) Aarabi, M., & Hasanian, S., 2014. Capacity planning and control: a review. Int. Journal of Scientific & Engineering Research. Vol. 5, Issue 8: pp. 975-984.

* [[7]](#Pierskalla_texto) Pierskalla W. P, & Brailer D., 1994. Application of operations research in health care delivery. In: Pollock S, Barnett A, Rothkopf M (eds) Beyond the profit motive: public sector applications and methodology. Handbooks in OR&MS, vol 6. North-Holland, New York.

* [[8]](#pardede_texto)  Pardede, A. (2019, 30 junio). LIMITED RESOURCES OPTIMIZATION OF HEALTH CARE SERVICES WITH A LINEAR INTEGER PROGRAMMING APPROACH. Research gate. <https://www.researchgate.net/publication/334130304_LIMITED_RESOURCES_OPTIMIZATION_OF_HEALTH_CARE_SERVICES_WITH_A_LINEAR_INTEGER_PROGRAMMING_APPROACH>

* [[9]](#jack_texto) Jack E. P., & Powers T L, 2009. A review and synthesis of demand and m

* [[10]](#Sitepu_texto) Sitepu, S., Mawengkang, H., & Husein, I. (2018). Optimization Model for Capacity Management and Bed Scheduling for Hospital. *IOP Conference Series: Materials Science and Engineering*, *300*, 012016. <https://doi.org/10.1088/1757-899x/300/1/012016>

* [[11]](#Vanjauten_texto) Hans, E. W., van Houdenhoven, M., & Hulshof, P. J., 2012. A framework for healthcare planning and control. Handbook of Healthcare System Scheduling (Springer).

* [[12]](#Wang_texto) Wang Y, Lee L H, Chew E P, Lam S S W, Low S K, Ong M E H, Li H. 2015 Multi-objective optimization for a hospital inpatient flow process via discrete event simulation. In Proceedings of the 2015 Winter Simulation Conference, edited by L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rosseti, 3622-3631. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc

* [[13]](#minsalud_texto)

* [[14]](#AMH_texto) A. M. H. Pardede et al., “Smart Health Model With A Linear Integer Programming Approach,” in Journal of Physics: Conference Series, 2019.
* [15] E. B. Nababan, A. R. Hamdan, S. Abdullah, and M. S. Zakaria, “A branch and bound algorithm in optimizing job shop scheduling problems,” in Proceedings - International Symposium on Information Technology 2008, ITSim, 2008.

* [[16]](#LZeng_texto) L. Gu, D. Zeng, S. Guo, A. Barnawi, and Y. Xiang, “Cost efficient resource management in fog computing supported medical cyber-physical system,” IEEE Trans. Emerg. Top. Comput., 2017.

* [[17]](#Dorling_texto) K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, “Vehicle Routing Problems for Drone Delivery,” IEEE Trans. Syst. Man, Cybern. Syst., 2017.

* [[18]](#Alsalemi_texto) A. Alsalemi et al., “Developing costeffective simulators for patient management: A modular approach,” in 2017 Fourth International Conference on Advances in Biomedical Engineering (ICABME), 2017, pp. 1–4.

* [[19]](#Tulus_texto) Tulus, I. Z. Sefnides, Sawaluddin, Suriati, and M. Dwiastuti, “Modeling of Sedimentation Process in Water,” in 2nd International Conference on Computing and Applied Informatics 2017, 2018, pp. 1– 5.

* [[20]](#Nasution_texto) M. K. M. Nasution, “Modelling and Simulation of Search Engine,” in International Conference on Computing and Applied Informatics 2016, 2017, pp. 1– 8.

* [[21]](#informacionCamas_texto) *Uso-UCIS-COVID19 | SALUDATA*. (s. f.). SALUDATA. Recuperado 21 de febrero de 2021, de https://saludata.saludcapital.gov.co/osb/index.php/datos-de-salud/enfermedades-trasmisibles/ocupacion-ucis/

* [[22]](#SanIgnacio_texto) Informe Gestion 2018 Gestion San Ignacio. (s. f.). Husi. Recuperado 21 de febrero de 2021, de <http://www.husi.org.co/documents/10180/14922653/1_+INFORME+DE+GESTION+HUSI+2018.pdf/d38ed0f5-d721-4d16-9c67-79d3faf215a8#:~:text=Cumplir%20con%20los%20excedentes%20operacionales,de%20la%20meta%20anual%20establecida>

* [[23]](#Tunal_texto) PRIETO VANEGAS, C. H. (2017, abril). INFORME DE GESTION ABRIL DE 2016 A MARZO DE 2017. SUBRED INTEGRADA DE SERVICIOS DE SALUD SUR E.S.E. <https://www.subredsur.gov.co/sites/default/files/control/INFORME%20DE%20GESTI%C3%93N%20Y%20RENDICI%C3%93N%20DE%20CUENTAS%20VIGENCIA%202016%20SUBRED%20SUR.pdf>

* [[24]](#SimonB_texto) El Nuevo Siglo. (2020, 31 enero). Inyección de $19.500 millones para el Simón Bolívar. https://www.elnuevosiglo.com.co/articulos/01-2020-inyeccion-de-19500-millones-para-el-simon-bolivar