Seminario
de
Proyecto Modelos Aplicados

Tema:

SexAppeal

En busca de una acto sexual ideal

Integrantes:
Dayron Fernández Acosta C311
Javier Villar Alonso C311
Julio José Horta C312
Daniel de la Cruz Prieto C311

No Institute Given

1. Introducción

Desde el inicio de los tiempos siempre ha habido la necesidad de que los dos sexos (mujer y hombre) se relacionen con la finalidad de que en un futuro se casen y así la especie pueda perdurar, lo cual es un instinto. El apego a sus allegados se demostraba en la consideración que tenían con los muertos y la decoración del acto fúnebre, y el amor propio se notaba en sus vestimentas además de la preocupación por sus ornamentas. Por eso se puede afirmar que fue la base para la creación y el surgimiento del amor para lo cual necesitaban de la comunicación.

El amor es una construcción cultural y cada período histórico ha desarrollado una concepción diferente del amor. Y es muy importante mencionar que el tipo de amor que se presenta durante las relaciones amorosas es el amor romántico es cual se define como una manifestación de atracción física, entre dos personas, como la afinidad compartida por dos individuos, también podríamos decir que el amor es un sentimiento que comparten dos personas aleatorias que se encuentran y no pueden evitar atraerse entre sí. A pesar de que las relaciones amorosas de los adolescentes no siempre han tenido el mismo significado, siempre han estado presentes, y no solo durante la adolescencia, sino también en las otras etapas de la vida humana pero en los tiempos actuales, la adolescencia es la etapa donde mayormente se generan los noviazgos y es también donde se centra la problemática de la investigación que se realiza.

En el momento que surge este interés amoroso, surge un fuerte deseo de actividad sexual entre la pareja inconscientemente, y a la vez una preocupación y a veces no logramos terminar como queremos una relación sexual.

En este trabajo abordaremos el problema de las relaciones sexuales y le daremos solución a diferentes modelos para que les sea más fácil realizar actos sexuales y puedan satisfacer siempre a su pareja

2. Problema Analizado

De un grupo que se reúne para realizar una actividad sexual nos interesa saber que secuencia para fomentar un resultado en específico, como lo es que el placer entre los individuos sea el máximo, o que el acto sexual dure el mayor tiempo posible, o hallar cual es el mejor resultado para que una cierta persona obtenga el mejor beneficio en ese acto sexual.

Para dar respuesta a este problema nos enfocamos en diversas variables:

posturas	→	listas de las posturas disponibles para el grupo de personas
personas	→	listas de las personas que realizarán el acto sexual
personaXplacer (PGUT)	→	placer que le otorga a cada persona por cada posturas
personaXenergia (ECUT)	→	energía que pierde cada persona por cada postura
enegiaInicial (EIP)	→	Energía inicial de la persona
placerInicial (PIP)	→	Placer inicial de la persona
placerRequerido (NPPOO)	→	Placer Requerido de la persona para llegar al orgasmo

Donde los modelos a estudiar serían los siguientes:

3. Modelos:

3.1. Maximizar la duración del acto sexual:

En este caso nos interesa extender la duración de un acto sexual, lo que implica realizar el mayor tiempo de posturas que logre satisfacer a las personas que anden realizando el acto sexual y que cumplan con los requisitos suficientes para que todos estén satisfechos, para ello es necesario que realizaremos un modelo donde nos interesa maximizar la sumatoria de los tiempos dedicados a cada postura.

Luego procedemos a trabajar las siguientes restricciones:
$1-)\ \sum_{pos} PGUT[per][pos]t[pos] >= NPPOO[per] - PIP[per]$

Esta restricción expresa que el placer alcanzado por cada persona en las sumas de las posturas ejecutadas debe ser mayor o igual a lo necesario para llegar al orgasmo

$2-)\ \sum_{pos} ECUT[per][pos] * t[pos] <= EIP[per]$

Esta restricción expresa que durante el acto sexual ninguna persona debe superar la energía que puede emplear en el acto sexual

3.2. Maximizar el placer del que menor placer alcance al finalizar el acto sexual:

Para este modelo nos interesa encontrar a la persona que menor placer pueda alcanzar y proceder a a maximizar su placer. Para ello maximizaremos una variable h que será el placer que alcance la persona h y mantendremos de variables los tiempos de cada postura.

Luego le añadiremos las siguientes restricciones:

$1-)\ \sum_{pos} PGUT[per][pos] * t[pos] >= h - PIP[per]$

En esta restricción expresa que la suma de los placeres alcanzados por las personas en el acto sexual debe de ser mayor o igual al placer alcanzado por el de menor placer menos la cantidad de placer necesaria para que alcance el orgasmo

$2-)\ \sum_{pos} t[pos] * PUGT[per][pos] >= NPPOO[p] - PIP[per]$

Esta restricción expresa que el placer alcanzado por cada persona en las sumas de las posturas ejecutadas debe ser mayor o igual a lo necesario para llegar al orgasmo

$3-)\ \sum_{pos} t[pos] * ECUT[per][pos] <= EIP[p]$

Esta restricción expresa que durante el acto sexual ninguna persona debe superar la energía que puede emplear en el acto sexual

3.3. Minimizar el cansancio del participante con mayor cansancio al analizar el acto sexual:

Este modelo es parecido al anterior, pero en este caso nos interesa minimizar el cansancio del que mayor cansancio alcanza al finalizar el acto sexual, la cual implica maximizar la energía de esta persona, para ello minimizaremos h y mantendremos como variables los tiempos de cada postura.

Luego añadiremos las siguientes restricciones:

$1-)\ \sum_{pos} ECUT[per][pos] * t[pos] >= h - EIP[per]$

Donde expresamos que todas las personas en el acto solo pueden gastar una energía mayor o igual a la diferencia entre la energía del que mayor cansancio acumula menos la energía inicial de la persona antes de comenzar el acto sexual

$2-)\ \sum_{pos} PGUT[per][pos] * t[pos] >= NPPOO[per] - PIP[per]$

Esta restricción expresa que el placer alcanzado por cada persona en las sumas de las posturas ejecutadas debe ser mayor o igual a lo necesario para llegar al orgasmo

$3-)\ \sum_{pos} t[pos] * ECUT[per][pos] <= EIP[p]$

Esta restricción expresa que durante el acto sexual ninguna persona debe superar la energía que puede emplear en el acto sexual

3.4. Minimizar la energía inicial de todos los participantes de forma que al terminar todos hayan alcanzado el orgasmo y tengan la misma energía:

En este modelo minimizaremos un h de forma que podamos igualar todas las energías de cada persona y llevaremos como variables tanto el tiempo como las energías de los participantes en el acto sexual.

Luego añadimos las siguientes restricciones:

$1-)\ \sum_{pos} ECUT[per][pos] * t[pos] == EIP[per] - h$

Donde buscamos un h tal que se igualen los EIP a la cantidad de energía gastada por las personas involucradas en el acto sexual. Siendo EIP variables no definidas. Además sin notan, al despejar el valor de EIP podemos verificar que el EIP de cada persona diferenciado con la energía gastada en el acto sexual por dicha persona, se volveran iguales a dicho h, lo que quiere decir que cada EIP - ECUT seran iguales el resultado para cada persona

$2-)\ \sum_{pos} ECUT[per][pos] * t[pos] <= EIP[per]$

Esta restricción expresa que durante el acto sexual ninguna persona debe superar la energía que puede emplear en el acto sexual

$3-)\ \sum_{pos} PGUT[per][pos] * t[pos] >= NPPOO[per] - PIP[per]$

Esta restricción expresa que el placer alcanzado por cada persona en las sumas de las posturas ejecutadas debe ser mayor o igual a lo necesario para llegar al orgasmo

3.5. Maximizar el placer inicial de un participante específico, de forma tal que todos los participantes, excepto el específico, alcancen el orgasmo:

Para este ejercicio trabajamos maximizando un h que indicará el placer de la persona definida por el usuario para maximizar su placer sin que llegue al orgasmo, y con los tiempos como variables del problema y añadimos de restricciones las siguientes:

$1-)\ \sum_{pos} ECUT[per][pos] * t[pos] <= EIP[per]$

Donde buscamos que la energía gastada por las personas en el acto sexual para hacer las posturas sea menor o igual que la energía que tenía a disposición la persona para llevar a cabo toda la actividad

$2-)\ \sum_{pos} PGUT[per][pos] * t[pos] >= NPPOO[per] - PIP[per]$

Esta restricción expresa que el placer alcanzado por cada persona en las sumas de las posturas ejecutadas debe ser mayor o igual a lo necesario para llegar al orgasmo

3−) $\sum_{pos} hPGUT[pos] * t[pos] < hNPPOO$

En esta restricción nos interesa que la persona definida en el modelo no alcance el orgasmo, por ello la cantidad de placer obtenida en el acto debe ser menor al placer requerido

4−) $hNPPOO > h$

En esta restricción buscamos maximizar el placer de h pero imponiendo que no pueda superar el placer requerido para llegar al orgasmo

5−) $\sum_{pos} PGUT[per][pos] * t[pos] >= NPPOO[per] - PIP[per]$

Esta restricción expresa que el placer alcanzado por cada persona en las sumas de las posturas ejecutadas debe ser mayor o igual a lo necesario para llegar al orgasmo

4. Programación los modelos

En este trabajo usamos la biblioteca pulp, un repo muy usado en la optimización de modelos que nos facilitará hallar los resultados de nuestro problema. Esta implica en la construccion del modelo mediante clasificaciones y agregándolo a una variable que nos permitirá procesar el problema

PuLP permite indicar el tipo de problema que hay que optimizar mediante palabras reservadas de la propia librería, maximización (LpMaximize) o minimización (LpMinimize), que deberán usarse cuando comenzamos a definirlo. Además, incluye soporte base para todos y cada uno de los elementos básicos de un problema de optimización:

- Variables (LpVariable): la cual usaremos para declarar las variables de nuestro modelo
- Función objetivo
- Restricciones o constraints

4.1. Código de los modelos:

```
import pulp as pl

#Instanciando variables
(...)

problem = pl.LpProblem("Maximizar el tiempo",pl.LpMaximize)

# Creando el Modelo
x_name = ['time1', 'time2']
x = [pl.LpVariable(x_name[i] , lowBound=0, upBound=200) for i in range(len(x_name))]
c = pl.LpAffineExpression([(x[i],1)  for i in range(len(x))])

# Instanciando el problema
problem += pl.lpSum([1 * x[i] for i in range(len(x_name))])

#Añadiendo Restricciones al problema
for person in range (len(personas)):
  problem += pl.lpSum(personaXplacer[person][i] * x[i] for i in range(len(posturas))) >=
placerRequerido[person] - placerInicial[person] , "placer para : " +
str(personas[person])


for person in range(len(personas)):
  problem += pl.lpSum(-personaXenergia[person][i] * x[i] for i in range(len(posturas)))
>= -enegiaInicial[person] , " energia para la persona : " + str(personas[person])

print(problem)

problem.solve()
```

Dicho código representa el primer modelo de nuestro problema y su forma de trabajar es sencilla. Primero instanciamos las variables fijas que queremos tener para luego plantear el tipo de modelo que queremos efectuar, ya sea de minimización o maximización usando la sintaxis LpMinimize o LpMaximize a la hora de instanciar el LpProblem que es donde guardaremos todo nuestro modelo

Luego para este caso procedemos a instanciar el problema que debemos resolver la cual en este caso buscamos maximizar la suma de los tiempos de un acto sexual

Luego seguiría instanciar nuevas variables en el caso que no solo se quieran las variables del modelo que queremos proceder a calcular usando LpVariable pero para este caso no es necesario y solo pasaremos al siguiente paso

En el siguiente paso vamos agregando poco a poco al problema las restricciones que definen al modelo del problema

Ya luego de haber seguido estos pasos podemos mandar a imprimir sus valores y enviar el resultado obtenido

```
problem = pl.LpProblem("Maximizar el placer H",pl.LpMaximize)
    h = pl.LpVariable("H",lowBound=0,cat=pl.LpInteger)
    problem+= h


    TimepositionVars=[]
    for position in range(len(Positions)):

 TimepositionVars.append(pl.LpVariable(Positions[position],lowBound=1,cat=pl.LpInteger))

    #por cada persona
    for person in range (len(Persons)):
        #Por cada tiempoXpostura x placerXpostura sumalos y revisa que sean mayores que
una variable de decision h menos el placer inicial
        problem += pl.lpSum(TimepositionVars[position]*PGUT[person][position] for
position in range(len(Positions)))  >=h- PIP[person], "placer para : " +
str(Persons[person])
        #Por cada tiempoXpostura x placerXpostura sumalos y revisa que sean mayores que
el placer necesario para el orgasmo
        problem += pl.lpSum(TimepositionVars[position]*PGUT[person][position] for
position in range(len(Positions))) >=  NPPOO[person] - PIP[person], "placer máximo de :
"+ str(Persons[person])
        #Por cada tiempoXpostura x cansancioXpostura sumalos y revisa que sean menores
que la energia inicial de la persona
        problem += pl.lpSum( TimepositionVars[position]*ECUT[person][position] for
position in range(len(Positions))) <= EIP[person], "Energía de : "+ str(Persons[person])

    print(problem)
```

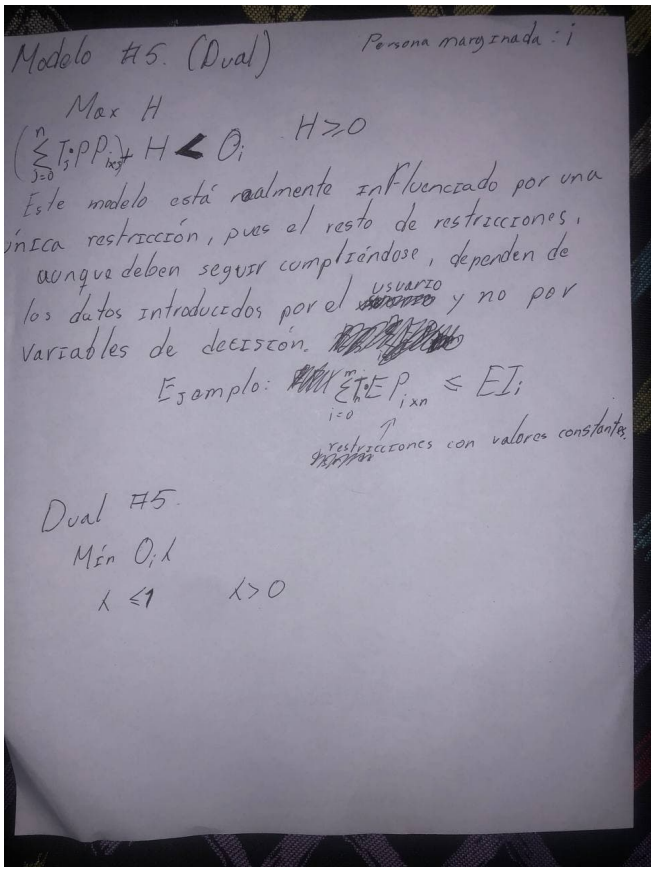
En este modelo que radica en lo que hablamos del modelo 2 podemos ver que la forma de proceder que explicamos en la anterior imágenes es similar, lo unico que ya aqu'i usaremos las LpVariables que planteamos anteriormente, la cual nos sirve para plantear variables fuera del modelo principal

Para los próximos modelos los programaríamos de manera muy similar

5. Modelos Duales

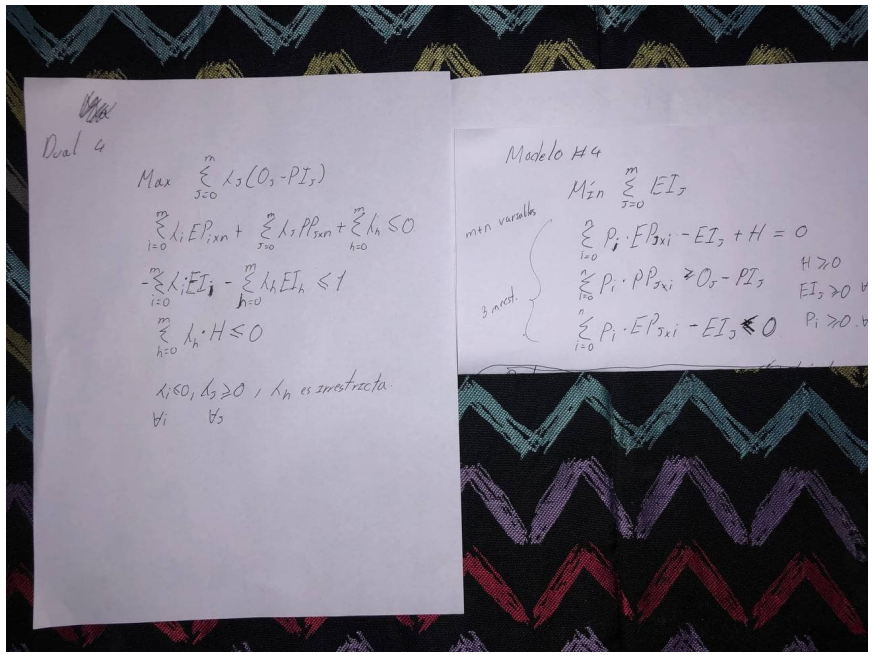
Ya como conocemos los modelos originales entonces para hallar los modelos duales solo es necesario modificar modelos y hacer transpuesta para poder hallar los modelos duales, ya que el procedimiento de calcularlos es mediante hallar pares de números que al multiplicar las restricciones se conviertan en la menor cota que odamos comparar con nuestra función modelo

Dicho esto podemos proceder a ver nuestros modelos duales:



6. Aplicación Presentada

La aplicación consiste en una página web hecha con streamlit donde nos permitirá proceder a realizar estos modelos agregando datos a nuestra conveniencia y obtener un resultado de acuerdo a la consulta mandada por el usuario



Dual 4

$$\text{Max} \sum_{j=0}^{n} \lambda_j (O_j - PI_j)$$

$$\sum_{i=0}^{m} \lambda_i EP_{ixn} + \sum_{j=0}^{n} \lambda_j PP_{jxn} + \sum_{h=0}^{m} \lambda_h \leq 0$$

$$-\sum_{i=0}^{m} \lambda_i EI_i - \sum_{h=0}^{m} \lambda_h EI_h \leq 1$$

$$\sum_{h=0}^{m} \lambda_h \cdot H \leq 0$$

$\lambda_i \leq 0$, $\lambda_j \geq 0$, λ_h es irrestricta.
$\forall i \quad \forall j$

Modelo #4

$$\text{Min} \sum_{j=0}^{m} EI_j$$

m+n variables

$$\sum_{i=0}^{n} P_i \cdot EP_{jxi} - EI_j + H = 0$$

$$\sum_{i=0}^{n} P_i \cdot PP_{jxi} \geq O_j - PI_j$$

$$\sum_{i=0}^{n} P_i \cdot EP_{jxi} - EI_j \leq 0$$

3 restr.

$H \geq 0$

$EI_j \geq 0 \ \forall$

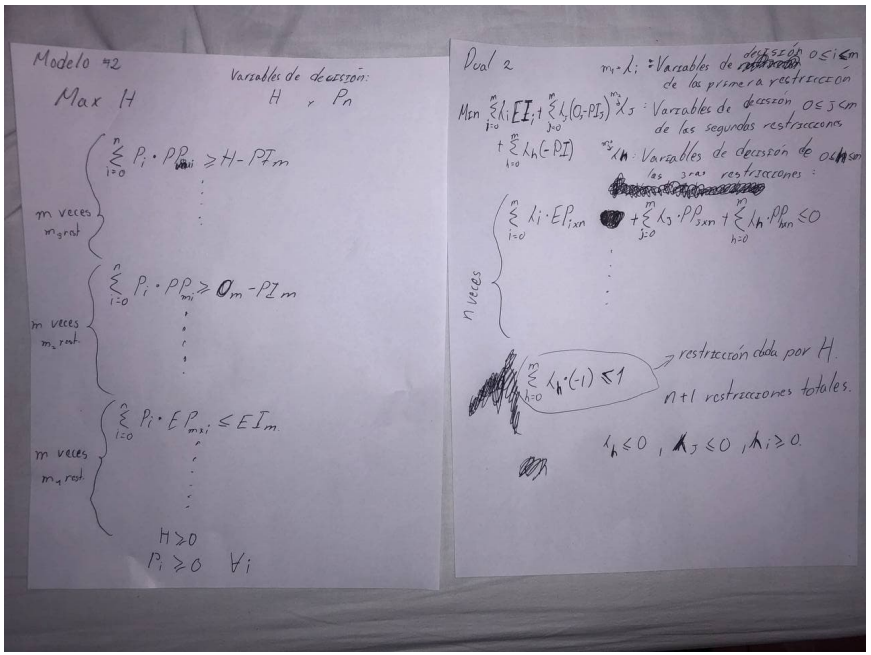
$P_i \geq 0 . \forall$

Dual # 3

Todo lo del dual del 3 es exactamente igual que el 2.

La única restricción que cambia es la que está circulada: (la restricción dada por H).

$$\sum_{j=0}^{n} k_j \leq 1$$



Modelo #2

Variables de decisión: H , P_n

Max H

$$\sum_{i=0}^{n} P_i \cdot PP_{mi} \geq H - PI_m$$

m veces m_1 rest.

$$\sum_{i=0}^{n} P_i \cdot PP_{mi} \geq 0_m - PI_m$$

m veces m_2 rest.

$$\sum_{i=0}^{n} P_i \cdot EP_{mxi} \leq EI_m$$

m veces m_3 rest.

$H \geq 0$

$P_i \geq 0 \quad \forall i$

Dual 2

$m_1 \cdot \lambda_i$: Variables de decisión $0 \leq i \leq m$ de la primera restricción

$\min \sum_{i=0}^{m} \lambda_i EI_i + \sum_{j=0}^{m} \lambda_j (0 - PI_j) + \sum_{h=0}^{m} \lambda_h (-PI)$

$m_2 \cdot \lambda_j$: Variables de decisión $0 \leq j \leq m$ de las segundas restricciones

$m_3 \cdot \lambda_h$: Variables de decisión de $0 \leq h \leq m$ las 3ras restricciones

$$\sum_{i=0}^{n} \lambda_i \cdot EP_{ixn} + \sum_{j=0}^{m} \lambda_j \cdot PP_{jxn} + \sum_{h=0}^{m} \lambda_h \cdot PP_{hxn} \leq 0$$

n veces

$\sum_{h=0}^{m} \lambda_h \cdot (-1) \leq 1$ → restricción dada por H.

$n+1$ restricciones totales.

$\lambda_h \leq 0 \ , \ \lambda_j \leq 0 \ , \ \lambda_i \geq 0.$

Modelo #1:

Variables:

Variable de decisión:

m: cantidad de personas.
n: cantidad de posturas.
P_n: matriz de [illegible] x postura.
PI_m: matriz de Placer Intecal
PP_{mxn}: matriz de Placer por Postura
EP_{mxn}: matriz de Energía por Postura.
O_m: matriz de Placer necesario para el orgasmo
EI_n: Energía intecal.

$$\text{Max} \sum_{i=0}^{n} P_i$$

m restricciones de cada manera

$$\left(\sum_{i=0}^{n} P_i \cdot PP_{jx1}\right) \geq O_j - PI_j \quad \text{donde } j \text{ es la persona } 0 \leq j < m$$

$$\left(\sum_{i=0}^{n} P_i \cdot PP_{mxi}\right) \geq O_m - PI_m$$

m restricciones

$$\left(\sum_{i=0}^{n} P_i \cdot EP_{jxi}\right) \leq E_j \quad \text{donde } j \text{ es } 0 \leq j < m$$

$$\left(\sum_{i=0}^{n} P_i \cdot EP_{mxi}\right) \leq E_m$$

$$P_i \geq 0 \quad \forall i$$

Dual #1: λ_i: variables del dual $0 \leq i <$ [illegible] de las primeras restricciones

λ_j: variables del dual [illegible] $\leq j <$ [illegible] de las [illegible] segundas restricciones

[illegible]

$$\text{Mín} \sum_{i=0}^{n} \lambda_i \cdot EI_i + \sum_{j=0}^{m} \lambda_j \cdot (O_j - PI_j)$$

n restricciones

$$\left(\sum_{i=0}^{n} \lambda_i \, EP_{ix0}\right) + \left(\sum_{j=0}^{m} \lambda_j \, PP_{jx0}\right) \leq 1$$

$$\left(\sum_{i=0}^{n} \lambda_i \, EP_{ixn}\right) + \left(\sum_{j=0}^{m} \lambda_j \, PP_{jxn}\right) \leq 1$$

$\lambda_i \geq 0$, $\lambda_j \leq 0$
$\forall i$ $\forall j$

6.1. Guía técnica de la aplicación

Para uso de la aplicación es necesario tener instalados los siguientes requerimientos:

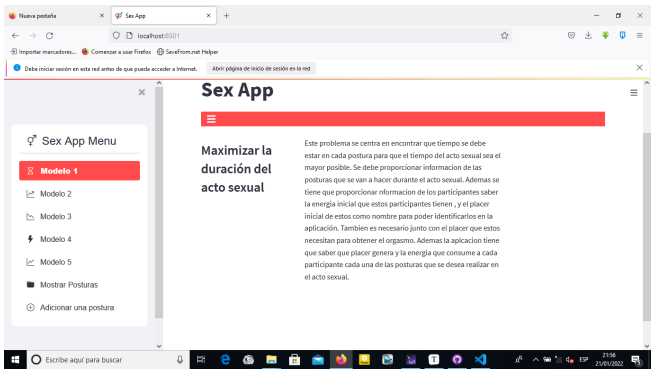
beautifulsoup4 == 4.10.0
numpy == 1.22.1
pandas == 1.3.5
PuLP == 2.6.0
scipy == 1.7.3
streamlit == 1.4.0

streamlit-option-menu

hydrali_components

Luego de todo esto iniciaremos por consola cmd desde la carpeta de la aplicación con el comando streamlit run sexapp.py

Una vez hecho deberá aparecer la siguiente pagina web:



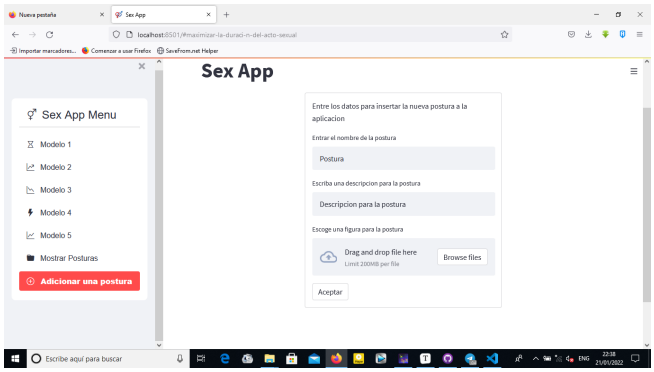
Esta página web es nuestra aplicación que consiste en un formulario en la que elegirás los datos a conveniencia y rellenarás información necesaria para poder operar la aplicación.

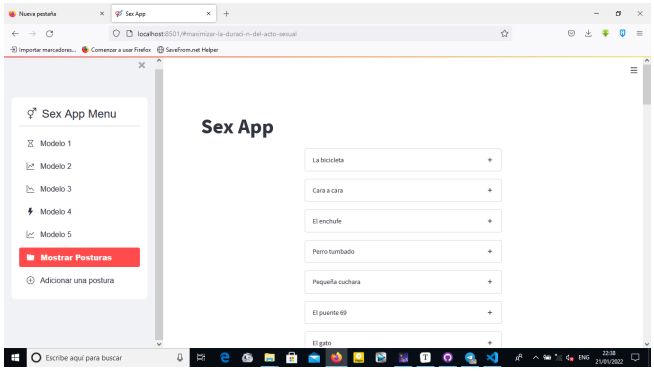
Para eso primero que nada podemos ver que en la barra izquierda podemos seleccionar un grupo de opciones donde 5 de ellos son los modelos a disposición para la aplicación, una opción para ver las posturas a disposición y otra opción para crear nuevas posturas que queramos usar en nuestra aplicación:

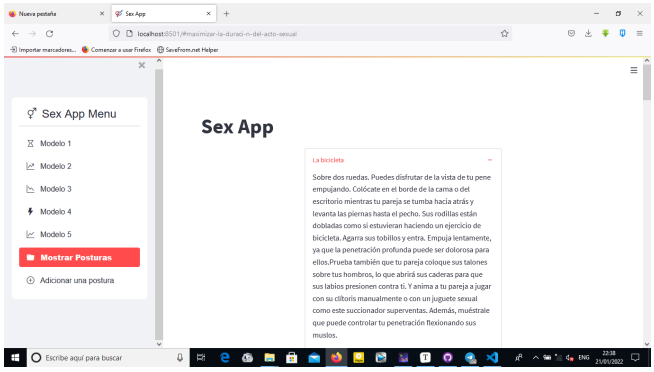
Volvamos a modelos1:

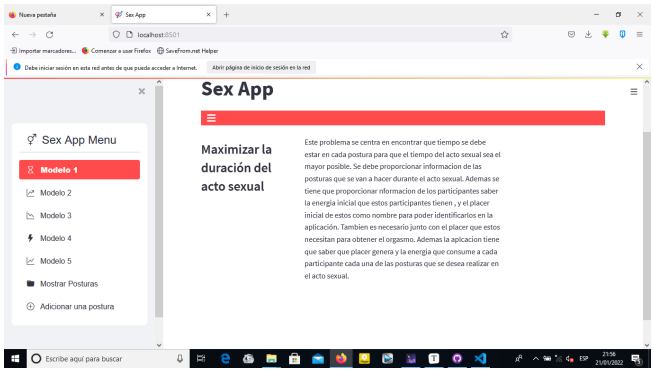
Si nos percatamos también nos sale una barra en la parte superior que cuando la expandimos nos salen un grupo de opciones:

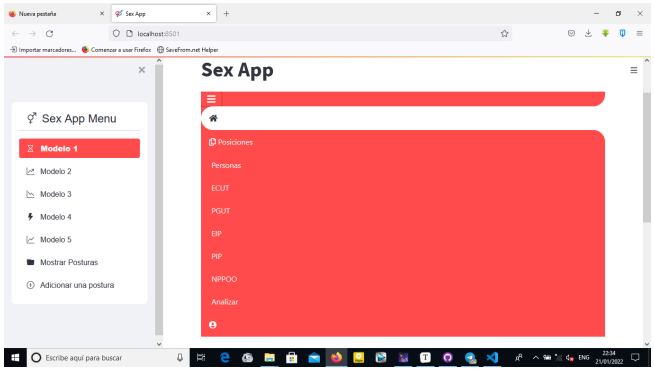
Si vamos a personas podemos agregar participantes para el acto sexual:

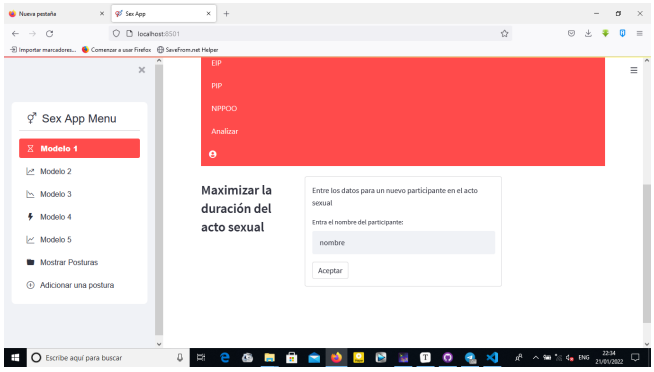

Sex App
Sex App Menu
Modelo 1
Modelo 2
Modelo 3
Modelo 4
Modelo 5
Mostrar Posturas
Adicionar una postura
Entre los datos para insertar la nueva postura a la aplicacion
Entrar el nombre de la postura
Postura
Escriba una descripcion para la postura
Descripcion para la postura
Escoge una figura para la postura
Drag and drop file here
Limit 200MB per file
Browse files
Aceptar

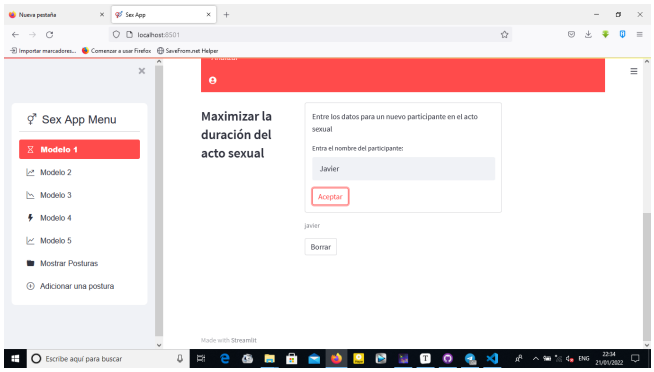

Sex App
Sex App Menu
Modelo 1
Modelo 2
Modelo 3
Modelo 4
Modelo 5
Mostrar Posturas
Adicionar una postura
La bicicleta
Cara a cara
El enchufe
Perro tumbado
Pequeña cuchara
El puente 69
El gato


Sex App
Sex App Menu
Modelo 1
Modelo 2
Modelo 3
Modelo 4
Modelo 5
Mostrar Posturas
Adicionar una postura
La bicicleta
Sobre dos ruedas. Puedes disfrutar de la vista de tu pene empujando. Colócate en el borde de la cama o del escritorio mientras tu pareja se tumba hacia atrás y levanta las piernas hasta el pecho. Sus rodillas están dobladas como si estuvieran haciendo un ejercicio de bicicleta. Agarra sus tobillos y entra. Empuja lentamente, ya que la penetración profunda puede ser dolorosa para ellos.Prueba también que tu pareja coloque sus talones sobre tus hombros, lo que abrirá sus caderas para que sus labios presionen contra ti. Y anima a tu pareja a jugar con su clítoris manualmente o con un juguete sexual como este succionador superventas. Además, muéstrale que puede controlar tu penetración flexionando sus muslos.

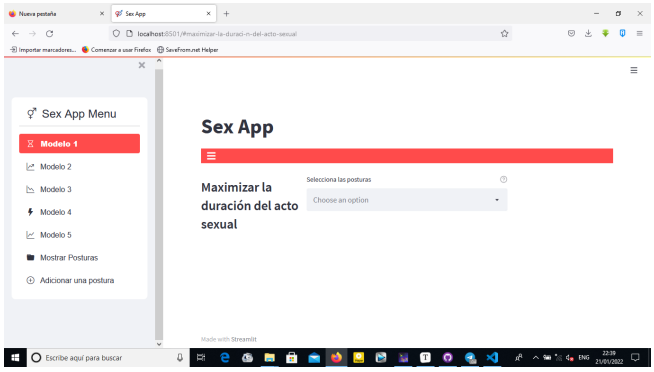

Sex App
Sex App Menu
Modelo 1
Modelo 2
Modelo 3
Modelo 4
Modelo 5
Mostrar Posturas
Adicionar una postura
Maximizar la duración del acto sexual
Este problema se centra en encontrar que tiempo se debe estar en cada postura para que el tiempo del acto sexual sea el mayor posible. Se debe proporcionar informacion de las posturas que se van a hacer durante el acto sexual. Ademas se tiene que proporcionar nformacion de los participantes saber la energia inicial que estos participantes tienen , y el placer inicial de estos como nombre para poder identificarlos en la aplicación. Tambien es necesario junto con el placer que estos necesitan para obtener el orgasmo. Ademas la aplicacion tiene que saber que placer genera y la energia que consume a cada participante cada una de las posturas que se desea realizar en el acto sexual.


Sex App
Sex App Menu
Modelo 1
Modelo 2
Modelo 3
Modelo 4
Modelo 5
Mostrar Posturas
Adicionar una postura
Posiciones
Personas
ECUT
PGUT
EIP
PIP
NPPOO
Analizar


Sex App
Sex App Menu
Modelo 1
Modelo 2
Modelo 3
Modelo 4
Modelo 5
Mostrar Posturas
Adicionar una postura
EIP
PIP
NPPOO
Analizar
Maximizar la duración del acto sexual
Entre los datos para un nuevo participante en el acto sexual
Entra el nombre del participante:
nombre
Aceptar



Si vamos a posturas podemos seleccionar las posturas que están disponibles para el acto sexual de las que hemos guardado en la aplicación:



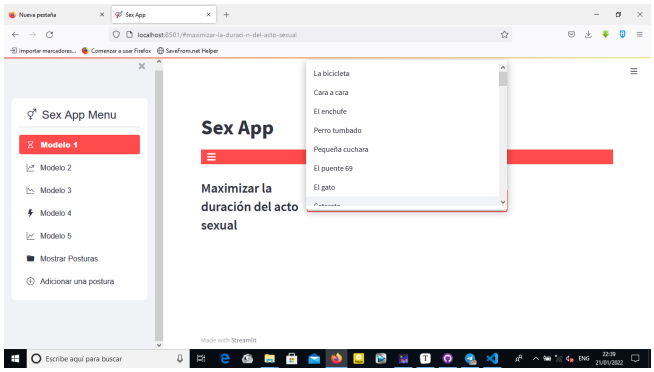
Si queremos eliminar de las elegidas alguna solo debemos tocar la X que sale en el nombre del que escogimos o agregamos (también es disponible dicha opción para personas)

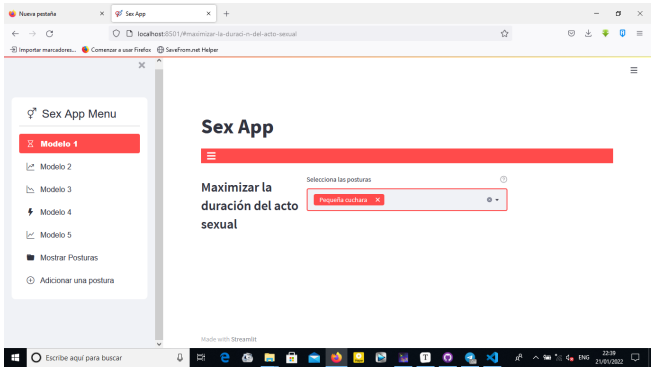
Si vamos ECUT, PGUT podemos ajustar sus datos de acuerdo a las posturas que añadimos al acto sexual disponibles para cada persona creada:

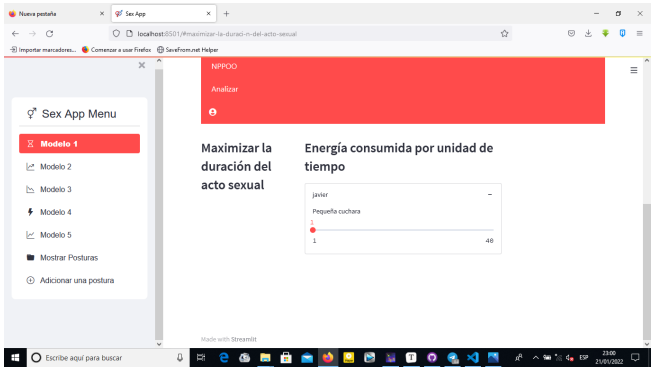
Si vamos a las opcione EIP, PIP y NPPOO podemos ajustar datos también de acuerdo a las personas que añadimos

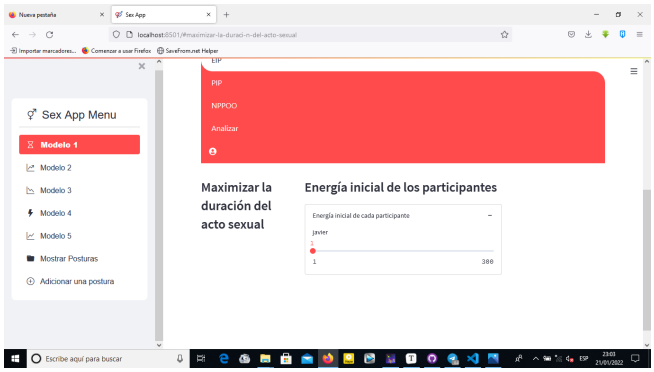
Además que cada una de esas 5 opciones planteadas anteriormente cuentan con una breve descripción del tipo de dato que describen en el modelo

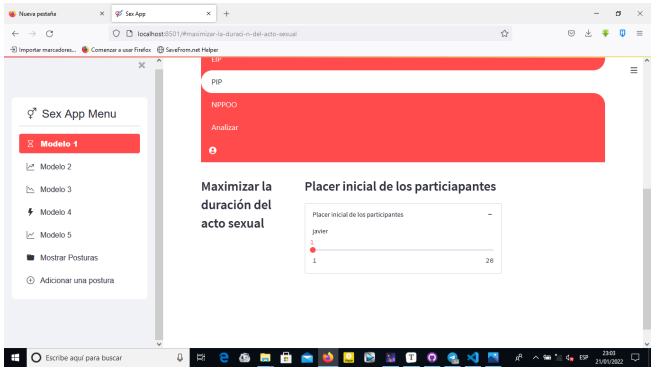
Una vez ingresado los datos queridos aparece una opción llamada .ªnalyce"que es para ejecutar nuestro problema, donde si encuentra un resultado válido a nuestros datos ingresados entonces imprimirá una serie de gráficos de comportamiento de nuestros datos:

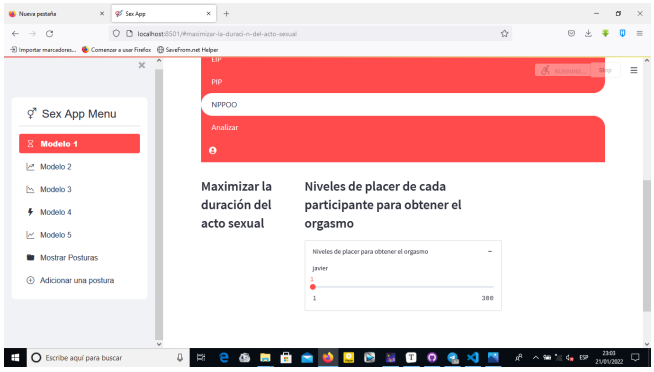

Sex App Menu
Modelo 1
Modelo 2
Modelo 3
Modelo 4
Modelo 5
Mostrar Posturas
Adicionar una postura
Sex App
Maximizar la duración del acto sexual
La bicicleta
Cara a cara
El enchufe
Perro tumbado
Pequeña cuchara
El puente 69
El gato


Sex App Menu
Modelo 1
Modelo 2
Modelo 3
Modelo 4
Modelo 5
Mostrar Posturas
Adicionar una postura
Sex App
Maximizar la duración del acto sexual
Selecciona las posturas
Pequeña cuchara


Sex App Menu
Modelo 1
Modelo 2
Modelo 3
Modelo 4
Modelo 5
Mostrar Posturas
Adicionar una postura
NPPOO
Analizar
Maximizar la duración del acto sexual
Energía consumida por unidad de tiempo
javier
Pequeña cuchara
1
40

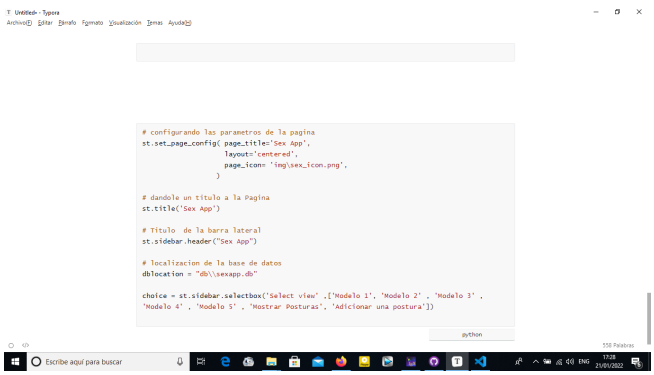

Sex App Menu
Modelo 1
Modelo 2
Modelo 3
Modelo 4
Modelo 5
Mostrar Posturas
Adicionar una postura
EIP
PIP
NPPOO
Analizar
Maximizar la duración del acto sexual
Energía inicial de los participantes
Energía inicial de cada participante
javier
1
380


Sex App Menu
Modelo 1
Modelo 2
Modelo 3
Modelo 4
Modelo 5
Mostrar Posturas
Adicionar una postura
EIP
PIP
NPPOO
Analizar
Maximizar la duración del acto sexual
Placer inicial de los particiapantes
Placer inicial de los participantes
javier
1
20


Sex App Menu
Modelo 1
Modelo 2
Modelo 3
Modelo 4
Modelo 5
Mostrar Posturas
Adicionar una postura
EIP
PIP
NPPOO
Analizar
Maximizar la duración del acto sexual
Niveles de placer de cada participante para obtener el orgasmo
Niveles de placer para obtener el orgasmo
javier
1
380

7. Guía de modificación externa del código

Como decíamos nuestra aplicación se basa en una página web en la que nosotros damos un encabezado a la pagina de la siguiente forma: Donde usamos el



setpage para poder construir nuestra página web ue tomaría como título, usando el método st.title, el nombre de ´Sex app´.

Luego construimos la barra que aparece en la ubicación izquierda de cuando abrimos la página web con st.sidebar.header para poner el nombre de la pagina web y guardar en una variable llamada choice el dato que hayamos escogido en un st.sidebar.selectorbox que es la barra donde seleccionamos un grupo de opciones como lo es Modelo1, Modelo2 y asi sucesivamente.

Una vez obtenido el valor escogido pasamos a preguntar que valor tiene escogido el choice de la siguiente manera:

Atendiendo a estas preguntas realizaremos uno de los métodos definidos para realizar la correspondiente elección, en el caso de haber elegido un modelo procederemos de la siguiente manera: Escribiremos el título de nuestro nuevo encabezado con el nombre del modelo que andamos estudiando y una breve des-

T. Untitled - Typora

Archivo(F) Editar Párrafo Formato Visualización Temas Ayuda(H)

```
if choice == 'Modelo 1':
  show_modelo_1()
elif choice == 'Modelo 2':
  show_modelo_2()
elif choice == 'Modelo 3':
    show_modelo_3()
elif choice == 'Modelo 4':
  show_modelo_4()
elif choice == 'Modelo 5':
  show_modelo_5()
elif choice == 'Adicionar una postura':
  addposture()
elif choice == 'Mostrar Posturas':
  show_all_postures()
```

542 Palabras

Escribe aquí para buscar

ENG 17:29 21/01/2022

T. Untitled - Typora

Archivo(F) Editar Párrafo Formato Visualización Temas Ayuda(H)

```
def show_modelo_1():

  st.title('Maximizar la duración del  acto sexual')

  st.write("""
  Este problema se centra en encontrar que tiempo se debe estar en cada
  postura  para que el tiempo del acto sexual sea el mayor posible. Se
  debe proporcionar información de las posturas que se van a hacer durante
  el acto sexual. Ademas se tiene que proporcionar nformacion de los participantes
  saber la energia inicial que estos participantes tienen , y el placer inicial de estos
  como nombre para poder identificarlos en la aplicación. Tambien es necesario
  junto con el placer que estos necesitan para obtener el orgasmo. Ademas la aplcacion
tiene
  que saber que placer genera y la energia que consume a cada participante cada una de
las posturas
  que se desea realizar en el acto sexual.
  """)

  optionsPositions = get_selected_postures()

  addparticipant()

  participants = st.session_state['persons']
```

python

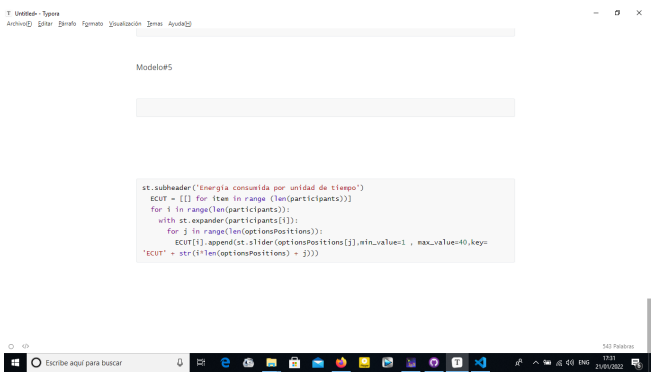
830 Palabras

Escribe aquí para buscar

ENG 17:30 21/01/2022

cripción de este modelo para luego llamar a un getselector que es el encargado de que el usuario pueda escoger las posturas que quiera analizar en este caso, luego llamaremos a un método participantes que es el formulario que aparece en la web, en la cual agregaremos los nombres qu escribamos y los guardaremos en participants

Una vez agregados los participantes podemos configurar los datos de cada uno de la siguiente manera: La idea siguiente es crear barra en el que adaptaremos a

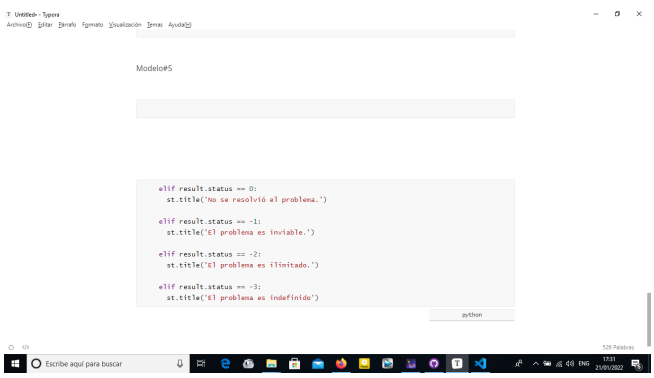


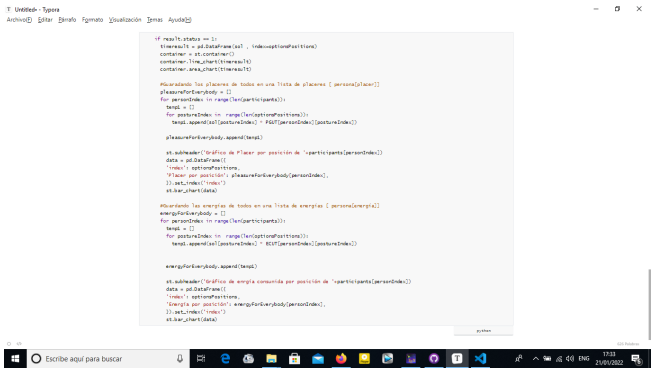
nuestra conveniencia el valor que creamos a conveniencia del dato que queramos guardar entre un intervalo para luego guardarla en uno de las variables que será la que represente dicho dato en el modelo que estamos trabajando

Este código lo repetiremos varias veces a la hora de introducir el EIP, el PGUT y demás datos que nos harán falta para resolver nuestros modelos

Luego de añadir todos los datos que queramos aparece en la página un botón llamado analyce, la cual lo usamos en nuestro código luego de la anterior imagen para cuando pulsen el botón empezar a revisar los posibles estados de resultados en nuestro modelo

Este último, cuando el estado sea 1, se encargará de imprimir una serie de gráficos vistos en el manual de usuario de la aplicación


T. Untitled - Typora
Archivo(F) Editar Párrafo Formato Visualización Temas Ayuda(H)
Modelo#5
elif result.status == 0:
st.title('No se resolvió el problema.')
elif result.status == -1:
st.title('El problema es inviable.')
elif result.status == -2:
st.title('El problema es ilimitado.')
elif result.status == -3:
st.title('El problema es indefinido')
python
526 Palabras
Escribe aquí para buscar
ENG
17:31
21/01/2022


T. Untitled - Typora
Archivo(F) Editar Párrafo Formato Visualización Temas Ayuda(H)
if result.status == 1:
timeresult = pd.DataFrame(asT , index=optionsPositions)
container = st.container()
container.line_chart(timeresult)
container.area_chart(timeresult)
#Guardando los placeres de todos en una lista de placeres [persona[placer]]
pleasureForEverybody = []
for personIndex in range(len(participants)):
temp1 = []
for postureIndex in range(len(optionsPositions)):
temp1.append(asT[postureIndex] * PLOT[personIndex][postureIndex])
pleasureForEverybody.append(temp1)
st.subheader('Gráfico de Placer por posición de '+participants[personIndex])
data = pd.DataFrame({
'index': optionsPositions,
'Placer por posición': pleasureForEverybody[personIndex],
}).set_index('index')
st.bar_chart(data)
#Guardando las energías de todos en una lista de energías [persona[energía]]
energyForEverybody = []
for personIndex in range(len(participants)):
temp1 = []
for postureIndex in range(len(optionsPositions)):
temp1.append(asT[postureIndex] * BEST[personIndex][postureIndex])
energyForEverybody.append(temp1)
st.subheader('Gráfico de energía consumida por posición de '+participants[personIndex])
data = pd.DataFrame({
'index': optionsPositions,
'Energía por posición': energyForEverybody[personIndex],
}).set_index('index')
st.bar_chart(data)
python
620 Palabras
Escribe aquí para buscar
ENG
17:33
21/01/2022

8. División organizativa del trabajo

En este trabajo cada uno trabajo ayudándose mutuamente sin dejar centrarnos en una tarea principal:

Daniel de la Cruz: realización de la aplicación por streamlit

Dayron Fernández: Análisis y programación de modelos, aportar ideas en la confección de la aplicación, confección de los duales

Julio José Horta: Análisis y programación de modelos, aportar ideas en la confección de la aplicación, confección de los duales

Javier Villar: Análisis de modelos, ayudar a cada uno de los integrantes en lo que le hiciera falta y confeccionar el Informe