



Instituto Tecnológico de Costa Rica

Escuela de Computación

IC6600 - Investigacion De Operaciones GR 60

Profesor: Jean Carlos Miranda Fajardo

Proyecto Parte #3 - Problemas de Mochila

Estudiantes:

Ricardo De Jesus Soto Araya - 2020035358

Brandon Guillermo Redondo Jimenez- 2019156567

Francisco González Madrigal - 2018107608

II Semestre - Sede de Limón

3 de octubre de 2022

Índice

Manual de usuario	3
Instrucciones de compilación.	3
Instrucciones de instalación	3
Instrucciones de ejecución	3
Uso de la aplicación: Abrir módulo de Problemas de Mochila	4
Descripción del problema	7
Solución del problema	8

Manual de usuario

Instrucciones de compilación.

Requisitos previos:

- Tener instalado NodeJS. Node.js (nodejs.org)
- Tener instalado SweetAlert2 SweetAlert

Instrucciones de instalación

- 1- Se descomprime el archivo .ZIP
- 2- Se abre una consola en la dirección de la carpeta descomprimida
- 3- Se escribe npm install para instalar los módulos necesarios de React (Se instalará sweetalert si no está previamente instalado.)

Instrucciones de ejecución

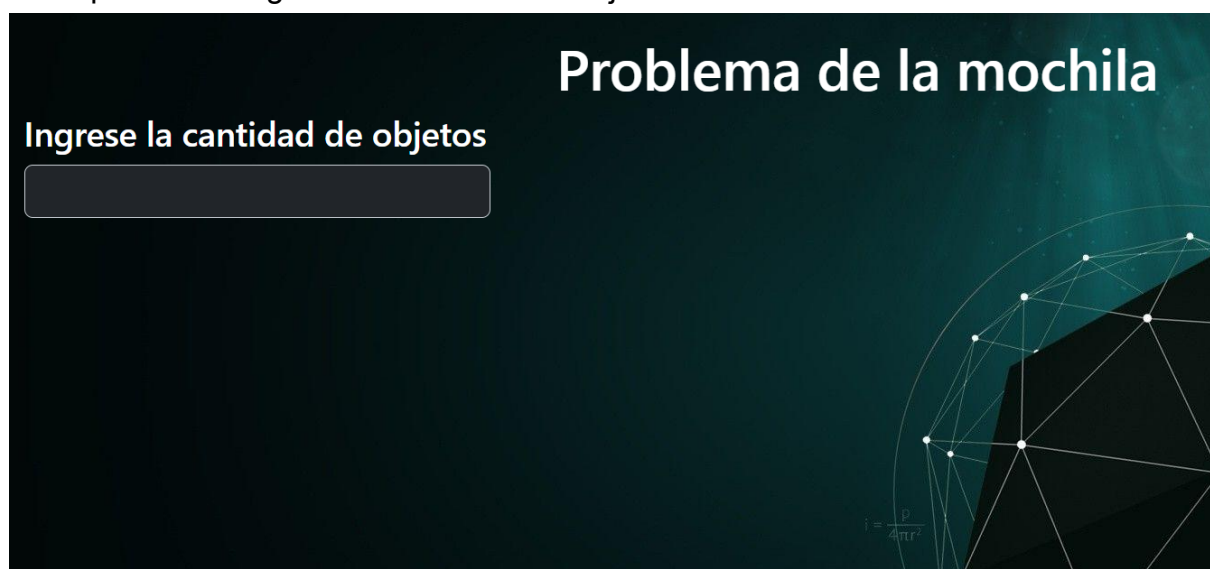
- 1- En consola desde la ubicación del proyecto -> npm start
- 2- Se procederá a mostrar la aplicación el navegador web predefinido automáticamente.

Uso de la aplicación: Abrir módulo de Problemas de Mochila

1- Se procederá a seleccionar la opción de Problemas de Mochila



2- Se procede a ingresar la cantidad de objetos



3- Se procede a llenar las tablas con la información respectiva y la cantidad total de la “mochila”

Problema de la mochila

Ingrese la cantidad de objetos

	Valor	Costo	Cantidad disponible
Objeto 1	<input type="text"/>	<input type="text"/>	<input type="text"/>
Objeto 2	<input type="text"/>	<input type="text"/>	<input type="text"/>
Objeto 3	<input type="text"/>	<input type="text"/>	<input type="text"/>
Objeto 4	<input type="text"/>	<input type="text"/>	<input type="text"/>

Ingrese la capacidad máxima

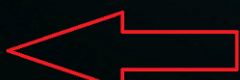
4- Al terminar de ingresar los datos se procede a seleccionar el botón de generar algoritmo (Se tomó como información el ejemplo del estudiante que entra a robar a una casa y hay solamente un objeto de cada tipo)

Ingrese la cantidad de objetos

	Valor	Costo	Cantidad disponible
Objeto 1	<input type="text" value="15000"/>	<input type="text" value="1"/>	<input type="text" value="1"/>
Objeto 2	<input type="text" value="5000"/>	<input type="text" value="4"/>	<input type="text" value="1"/>
Objeto 3	<input type="text" value="9000"/>	<input type="text" value="3"/>	<input type="text" value="1"/>
Objeto 4	<input type="text" value="10000"/>	<input type="text" value="5"/>	<input type="text" value="1"/>

Ingrese la capacidad máxima

Generar Algoritmo



5- Al clicar el botón se procederá a mostrar la resolución del problema tal y como se mostrará la siguiente información:

Maximizar:

$Z = 15000\text{obj1} + 5000\text{obj2} + 9000\text{obj3} + 10000\text{obj4}$

Sujeto a:

$1\text{obj1} + 4\text{obj2} + 3\text{obj3} + 5\text{obj4} \leq 8$

$\text{obj1} \leq 1$

$\text{obj2} \leq 1$

$\text{obj3} \leq 1$

$\text{obj4} \leq 1$

$\text{obji} \geq 0$

0	0	0	0	0
0	15000 , obj1 = 1	15000 , obj2 = 0	15000 , obj3 = 0	15000 , obj4 = 0
0	15000 , obj1 = 1	15000 , obj2 = 0	15000 , obj3 = 0	15000 , obj4 = 0
0	15000 , obj1 = 1	15000 , obj2 = 0	15000 , obj3 = 0	15000 , obj4 = 0
0	15000 , obj1 = 1	15000 , obj2 = 0	24000 , obj3 = 1	24000 , obj4 = 0
0	15000 , obj1 = 1	20000 , obj2 = 1	24000 , obj3 = 1	24000 , obj4 = 0
0	15000 , obj1 = 1	20000 , obj2 = 1	24000 , obj3 = 1	25000 , obj4 = 1
0	15000 , obj1 = 1	20000 , obj2 = 1	24000 , obj3 = 1	25000 , obj4 = 1
0	15000 , obj1 = 1	20000 , obj2 = 1	29000 , obj3 = 1	29000 , obj4 = 0

Resultado:

$\text{obj1} = 1$

$\text{obj2} = 1$

$\text{obj3} = 1$

$\text{obj4} = 0$

Esto indica que:

Se seleccionó: 1 Objeto1

Se seleccionó: 1 Objeto2

Se seleccionó: 1 Objeto3

El Objeto4 no se seleccionó

Valor máximo obtenido:29000

Descripción del problema

Se requiere crear una función dentro de la aplicación creada anteriormente en la segunda etapa del proyecto la cual es capaz de calcular la solución más óptima de selección de artículos, actividades, tareas..., etc por su respectivo valor, costo y capacidad total, por lo cual hay que programar en Javascript el código que servirá para solucionar el algoritmo anteriormente mencionado y así mostrarlo a su vez por múltiples componentes hechos en la librería de React.

Solución del problema

La solución realizada fue la utilización de múltiples componentes de React tanto funcionales como nativos de ES6, esto de forma que los nativos de ES6 son utilizados para realizar los cálculos del algoritmo, a su vez, los componentes funcionales fueron utilizados como componentes que serán reutilizables a lo largo de futuros cambios de la aplicación para así optimizar el tiempo de desarrollo y brindar más atención a la solución de los próximos algoritmos en lugar de programar como mostrarlos.

Con respecto al código utilizado para solucionar el algoritmo, lo que se procedió fue realizar un plan de exactamente qué procesos se realizan al elaborarlo a mano, después se procedió a realizar la codificación del mismo y finalmente la implementación del mismo en la parte visual de la aplicación para así llegar a la solución deseada y poderla mostrar al usuario por medio de los componentes anteriormente mencionados.

Análisis de resultados

Requerimiento	Estado	Observaciones
Ingresar los objetos, tareas, etc.	Completado	Funciona correctamente en cada una de sus funcionalidades.
Ingresar pesos de los objetos, tareas, etc.	Completado	Funciona correctamente en cada una de sus funcionalidades.
Ingresar capacidad máxima	Completado	Funciona correctamente en cada una de sus funcionalidades.
Despliegue de función de Maximizar	Completado	Funciona correctamente en cada una de sus funcionalidades.
Mostrar las reglas de "sujeto a"	Completado	Funciona correctamente en cada una de sus funcionalidades.
Mostrar todas tablas	Completado	Funciona correctamente en cada una de sus funcionalidades.
Mostrar el resultado	Completado	Funciona correctamente en cada una de sus funcionalidades.
Mostrar la cantidad específica de qué objetos llevar y cuales no	Completado	Funciona correctamente en cada una de sus funcionalidades.