



Sistema de control y prevención para disminuir el consumo irresponsable de energía eléctrica y emisiones GEI usando Estrategias de Programación Divide y Vencerás y Fuerza Bruta

ALGORITMOS Y COMPLEJIDAD

Julian Garcia Fabricio
Mendez Cruz Angely
Mendez Cruz Ciara

Realidad Problemática

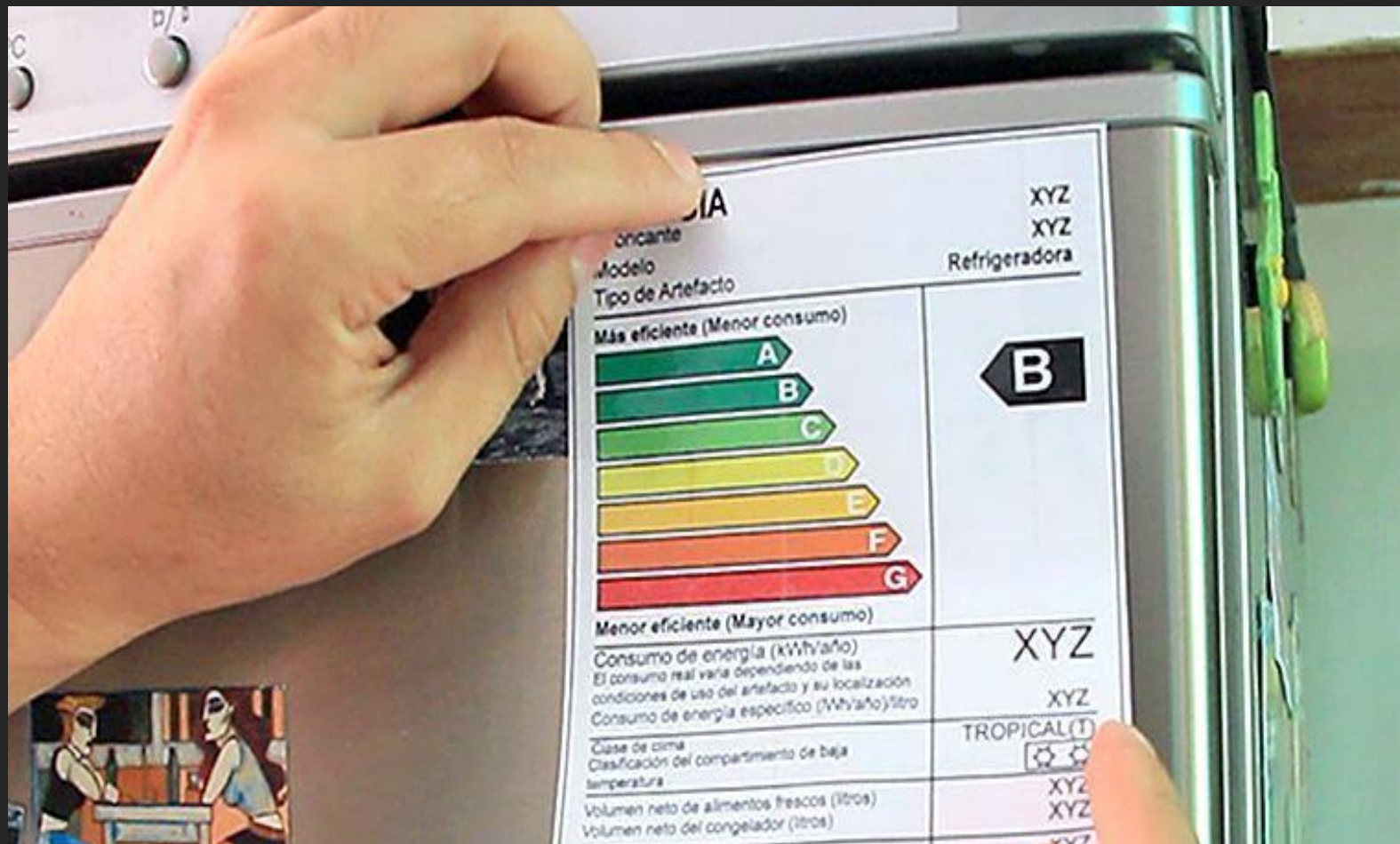
- Los problemas ambientales por los que el mundo atraviesa actualmente son realmente críticos y constituyen un desafío enorme.
- Diferentes documentos como Cambio Climático 2014, Informe de Síntesis del Panel Intergubernamental sobre Cambio Climático (IPCC) y demás, consolidaron el convencimiento de que las emisiones de GEI y sus consecuencias en el clima representan una externalidad negativa global.
- El Ministerio de Energía y Minas Peruano (MEM) informó que el impacto del uso de energía en los sectores doméstico, público, industrial, comercial y residencial, a nivel mundial, es responsable de más del 70% de la contaminación ambiental, por lo que es necesario una aplicación de políticas de eficiencia energética, fundamentales en la tarea de mitigar el daño ambiental.
- Por los antecedentes expuestos, nos planteamos la siguiente problemática ¿Será posible generar un sistema de control y prevención para disminuir el consumo irresponsable de energía eléctrica y emisiones GEI de los departamentos del Perú basados en los valores picos en Wh, usando las Estrategias de Programación Divide y Vencerás y Fuerza Bruta?



Consumo de electricidad de las familias en Lima creció 20% durante cuarentena

Justificación

La preocupación generada nos proporciona la importancia en señalar que se necesita un control y prevención adecuado para mitigar la contaminación del medio ambiente. Por lo que surge el propósito de disminuir el consumo irresponsable de energía eléctrica y emisiones de GEI.



MINEM recomienda a la población adquirir productos que tengan el Etiquetado de Eficiencia Energética.

Objetivos

General

- Desarrollar un sistema estándar de control y prevención usando las estrategias de programación divide y vencerás y fuerza bruta; con la finalidad de disminuir el consumo irresponsable de energía eléctrica y emisiones GEI en los departamentos del Perú, a través de una plataforma de gestión de información.

Específicos


- Emplear la estrategia de programación, fuerza bruta para la creación de la función que halle los puntos picos (más de un punto) de energía eléctrica producidos y emisiones de GEI, de los 25 departamentos del Perú, incluido Callao.
- Emplear también la estrategia de programación, divide y vencerás para la creación de otra función que permita obtener el máximo valor pico (solo uno) de la energía eléctrica producido y emisiones de GEI, de los 25 departamentos del Perú, incluido Callao

- Controlar y disminuir el consumo irresponsable de energía eléctrica, consecuentemente reducir las emisiones GEI en los departamentos del Perú, a través de la información adquirida y brindada a los usuarios por el programa.




Método

1. Investigación sobre la realidad problemática y los enfoques de las diversas soluciones. Planteándonos, las estrategias de Divide y Vencerás y Fuerza Bruta.



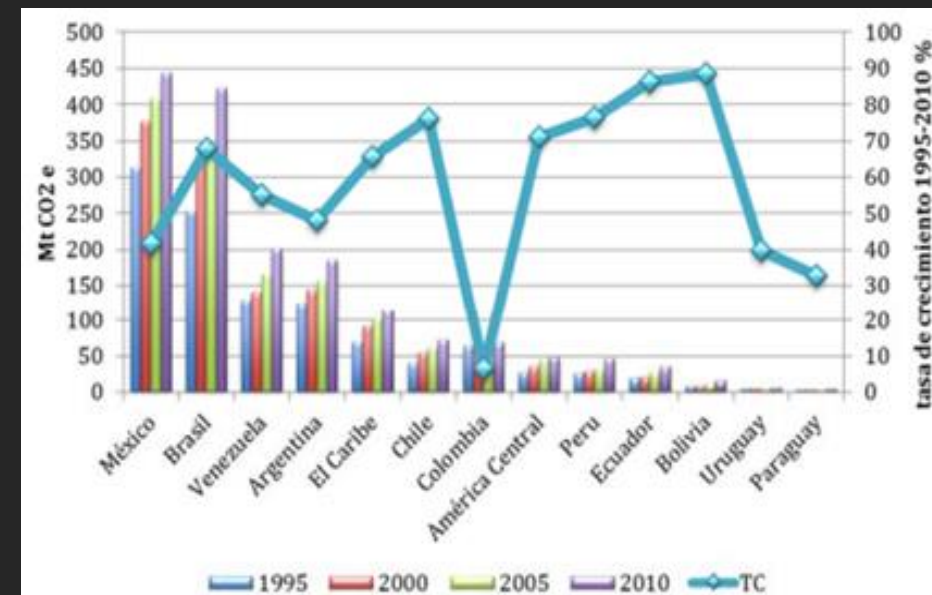
2. Registro de los valores de consumo de energía eléctrica de los 25 departamentos, incluido Callao, del mes de diciembre del 2021, según el Ministerio de Energía y Minas.



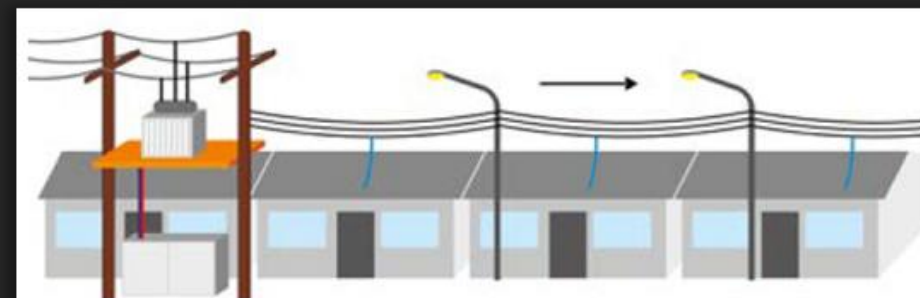
3. Elaboración de una base de datos que permitió identificar los departamentos con sus respectivos valores de consumo, análisis de la data generada, como el cálculo aproximado de las emisiones GEI.

Energía Eléctrica y Cambio Climático

- El cambio climático es consecuencia de una externalidad negativa global originada por las emisiones de gases de efecto invernadero (GEI) a la atmósfera.
- En el Perú, en el 2000 se consumía 357,774 TJ, y en 2017 se incrementó a 724,043 TJ; es decir, el consumo de energía aumentó en un 102% en 17 años.
- La energía entregada por el sistema de transmisión a las subestaciones de potencia permite el traslado de esta energía a los usuarios finales en baja tensión (220, 380 y 440 V), a través de las redes secundarias. El impacto ambiental en este tipo de actividad de distribución de energía eléctrica se focaliza en la etapa de operación, donde se hace uso del PCB, una de las sustancias más tóxicas que se puede encontrar en el ambiente.



Emisiones GEI totales en América Latina y el Caribe.
WRI (2015) .



Red Secundaria.

Estrategia Divide Y Vencerás

- Implica resolver un problema difícil, dividiéndolo en partes más simples tantas veces como sea necesario, hasta que la resolución de las partes se torna obvia. La solución del problema principal se construye con las soluciones encontradas.
- Se basa en la técnica de búsqueda binaria para verificar si el elemento medio es el elemento pico o no. Si el elemento medio no es el elemento pico, entonces siempre hay un elemento pico en el lado derecho. Y si el elemento del lado izquierdo es mayor que el elemento del medio, siempre hay un elemento de pico en el lado izquierdo.

```
1 | Función encuentra_PicoB (Parámetros: arr, bajo, alto, n)
2 | |   med <- bajo + (alto - bajo) / 2
3 | |   med <- Entero (med)
4 | |   Si ( ((med == 0) or (arr[med - 1] <= arr[med])) and ((med == n - 1) or (arr[med + 1] <= arr[med])) )
5 | | |   Retornar -> med
6 | | Fin Si
7 | | Sino Si ( (med > 0) and (arr[med - 1] > arr[med]) )
8 | | |   Retornar -> encuentra_PicoB (Parámetros: arr, bajo, (med - 1), n)
9 | | Fin Sino Si
10 | | Sino
11 | | |   Retornar -> encuentra_PicoB (Parámetros: arr, (med + 1), alto, n)
12 | | Fin Sino
13 | Fin Función encuentra_PicoB

14 | Función encuentra_PicoA (Parámetros: arr, n)
15 | |   Retornar -> encuentra_PicoB (Parámetros: arr, 0, n - 1, n)
16 | Fin Función encuentra_PicoA
```

Elaboración propia.

Complejidad del Algoritmo DYV

- Tenemos que n es el número de elementos que utilizará la función. En la función **encuentraPicoB**, una función recursiva, cada paso de búsqueda que se hace, no se utilizan completamente los n números, por lo que esta función se puede comparar a la búsqueda binaria, por lo que tendríamos una función de recurrencia de $T(n) = O(1) + T(n/2)$ cuya solución es: $O(\log n)$

```
1 | Función encuentra_PicoB (Parámetros: arr, bajo, alto, n)           //O (1)
2 | | med <- bajo + (alto - bajo) / 2                                //O (1)
3 | | med <- Entero (med)                                           //O (1)           //Función Si: O (1)
4 | | Si ( ((med == 0) o (arr[med - 1] <= arr[med])) and ((med == n - 1) o (arr[med + 1] <= arr[med])) )
5 | | | Retornar -> med                                           //O (1)
6 | | Fin Si
7 | | Sino Si ( (med > 0) and (arr[med - 1] > arr[med]) )           //O (1)
8 | | | Retornar -> encuentra_PicoB (Parámetros: arr, bajo, (med - 1), n) // O (Log n) Búsqueda Binaria
9 | | Fin Sino Si
10| | Sino                                                         //O (1)
11| | | Retornar -> encuentra_PicoB (Parámetros: arr, (med + 1), alto, n) //O (Log n) Búsqueda Binaria
12| | Fin Sino
13| Fin Función encuentra_PicoB                                     //O (Log n) Búsqueda Binaria

14| Función encuentra_PicoA (Parámetros: arr, n)                 //O (1)
15| | Retornar -> encuentra_PicoB (Parámetros: arr, 0, n - 1, n) //O (1)
16| Fin Función encuentra_PicoA                                     //O (1)
```

Elaboración propia.

Estrategia Fuerza Bruta

- Los algoritmos de fuerza bruta, son capaces de encontrar la solución a un problema por más complicado que sea. Su fundamento es simple, probar todas las posibles combinaciones, recorrer todos los posibles caminos hasta dar con la solución.
- El algoritmo consiste, si en el arreglo el primer elemento es mayor que el segundo o el último elemento es mayor que el penúltimo, añade dichos elementos al arreglo auxiliar. De lo contrario, atraviase el arreglo desde el segundo índice hasta el penúltimo índice.

```
1 | picos <- []
2 | Función encuentraPico (Parámetros: arr, n)
3 |   Si (n == 1)
4 |   |   Retornar -> picos -> Adjuntar (n - 1)           //picos.append (n - 1)
5 |   Fin Si
6 |   Si (arr[0] >= arr[1])
7 |   |   picos -> Adjuntar (0)                             //picos.append (0)
8 |   Fin Si
9 |   Para i = 0 Hasta n - 1 Entonces //O(n) Búsqueda lineal
10 |   |   Si (arr[i] >= arr[i - 1] and arr[i] >= arr[i + 1])
11 |   |   |   picos -> Adjuntar (i)                         //picos.append (i)
12 |   |   Fin Si
13 |   Fin Para
14 |   Si (arr[n - 1] >= arr[n - 2])
15 |   |   picos -> Adjuntar (n - 1)                         //picos.append (n - 1)
16 |   Fin Si
17 |   Retornar -> picos
18 | Fin Función encuentraPico
```

Elaboración propia.

Complejidad del Algoritmo FB

- Tenemos que n es el número de elementos que utilizará la función. En la función **encuentraPico** tenemos una búsqueda lineal factorial, por lo que se recorrerían $n - 1$ números, Tendríamos una función de recurrencia de $T(n - 1) + O(1)$, cuya complejidad es $O(n)$

```
1 | picos <- []
2 | Función encuentraPico (Parámetros: arr, n) //O (1)
3 | | Si (n == 1) //O (1)
4 | | | Retornar -> picos -> Adjuntar (n - 1) //O (1)
5 | | Fin Si
6 | | Si (arr[0] >= arr[1]) //O (1)
7 | | | picos -> Adjuntar (0) //O (1)
8 | | Fin Si
9 | | Para i = 0 Hasta n - 1 Entonces //O (n) Búsqueda lineal
10| | | Si (arr[i] >= arr[i - 1] and arr[i] >= arr[i + 1]) //O (1)
11| | | | picos -> Adjuntar (i) //O (1)
12| | | Fin Si
13| | Fin Para
14| | Si (arr[n - 1] >= arr[n - 2]) //O (1)
15| | | picos -> Adjuntar (n - 1) //O (1)
16| | Fin Si
17| | Retornar -> picos //O (1)
18| Fin Función encuentraPico
```

Elaboración propia.

Implementación



PYTHON



```
def encuentra_PicoB(arr, bajo, alto, n):
    #band = 0
    # Encuentra el índice del elemento medio
    med = bajo + (alto - bajo)/2
    med = int(med)

    # Compara el elemento del medio con su
    # vecinos (si existen vecinos)
    if ((med == 0 or arr[med - 1] <= arr[med]) and
        (med == n - 1 or arr[med + 1] <= arr[med])):
        return med

    # Si el elemento medio no es pico y
    # su vecino izquierdo es mayor
    # que eso, luego la mitad izquierda debe
    # tener un elemento pico
    #if band == 1:
    elif (med > 0 and arr[med - 1] > arr[med]):
        return encuentra_PicoB(arr, bajo, (med - 1), n)

    # Si el elemento medio no es pico y
    # su vecino derecho es mayor
    # que eso, entonces la mitad derecha debe
    # tener un elemento pico
    else:
        return encuentra_PicoB(arr, (med + 1), alto, n)
```

Función encuentra_PicoB con Divide Y
Vencerás

```
def encuentraPico(arr, n) :
    # Si solo hay elemento
    if (n == 1) :
        return picos.append(n-1)

    #Si el primer elemento es mayor que el segundo
    if (arr[0] >= arr[1]) :
        picos.append(0)

    #Comprobar todos los demás elementos
    for i in range(0, n-1) :
        # Comprobar si los vecinos son más pequeños
        if (arr[i] >= arr[i - 1] and arr[i] >= arr[i + 1]):
            picos.append(i)

    #Si el último elemento es mayor que el penúltimo
    if (arr[n - 1] >= arr[n - 2]) :
        picos.append(n-1)

    return picos
```

Función encuentraPico con Fuerza Bruta

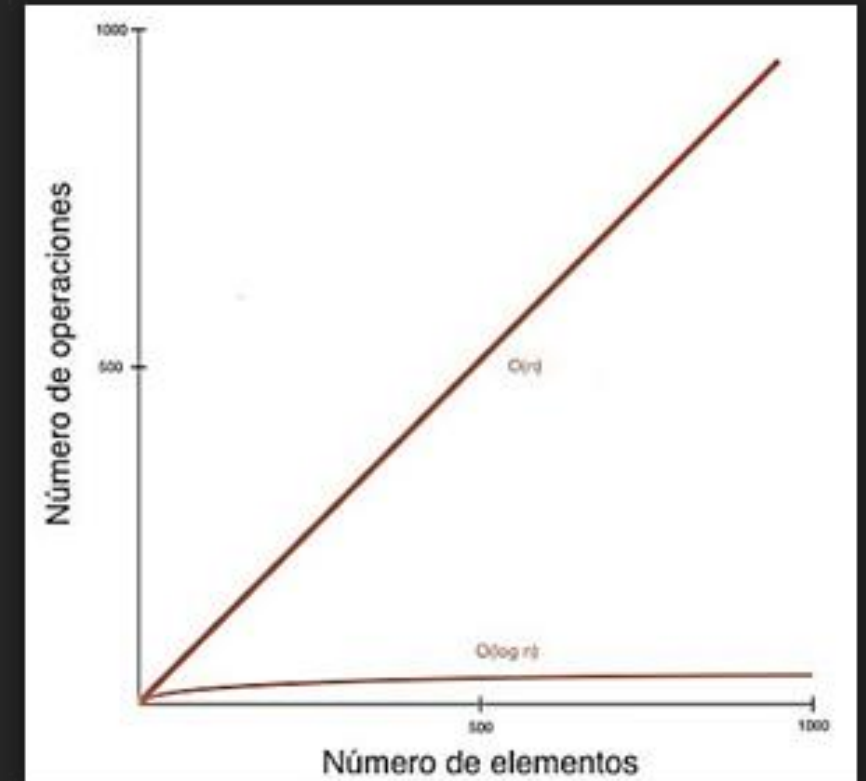
Resultados Teóricos

Para el algoritmo de **Fuerza Bruta** la complejidad fue $O(n)$, mientras que para el algoritmo **Divide y Vencerás** fue $O(\log n)$.

La complejidad de un algoritmo nos permite entender cómo se va a comportar el algoritmo cuando **incrementemos la cantidad de datos** de entrada, en esta investigación la cantidad de datos irá en aumento si usamos otra base de datos (más valores de energía eléctrica), por lo que es recomendable elegir un buen algoritmo.

Se recomienda usar Divide y Vencerás, si se necesita conocer solo 1 valor pico; porque usará **pocos recursos**, en el tiempo que lleve a ejecutarse y la cantidad de espacio de memoria que requerirá.

Por otro lado, **Fuerza Bruta** si se necesita conocer más de 1 valor pico; sin embargo empleará **más recursos**.



Complejidad Logarítmica VS Complejidad Lineal,
por GitBook.

Resultados Prácticos

Para la obtención de los valores de las emisiones GEI (GgCO₂eq), se realizó basado en los datos del Reporte Semestral de Monitoreo del Mercado Eléctrico del 2016 del Osinergrmin (Organismo Supervisor de la Inversión de Energía y Minería), que indicó que la **producción total de energía eléctrica** a nivel nacional totalizó **51,768 GWh**. Además, de lo mostrado por el Inventario Nacional De Gases De Efecto Invernadero (2016) del Ministerio del Ambiente Peruano, que menciona que el segundo sector con mayores **emisiones de GEI** reportada es Energía, con **58,132.54 GgCO₂eq**.

	Energía Eléctrica (GWh)		Emisiones GEI (GgCO ₂ eq)
2016	51,768		58,132.54
Diciembre 2021	X		Y

$$Y = \frac{X \text{ GWh} \cdot (58,132.54 \text{ GgCO}_2\text{eq})}{51,768 \text{ GWh}}$$

Los resultados de los algoritmos de Divide y Venceras y Fuerza Bruta sobre los valores picos son los siguientes:

Resultados Valores Pico: Diciembre 2021

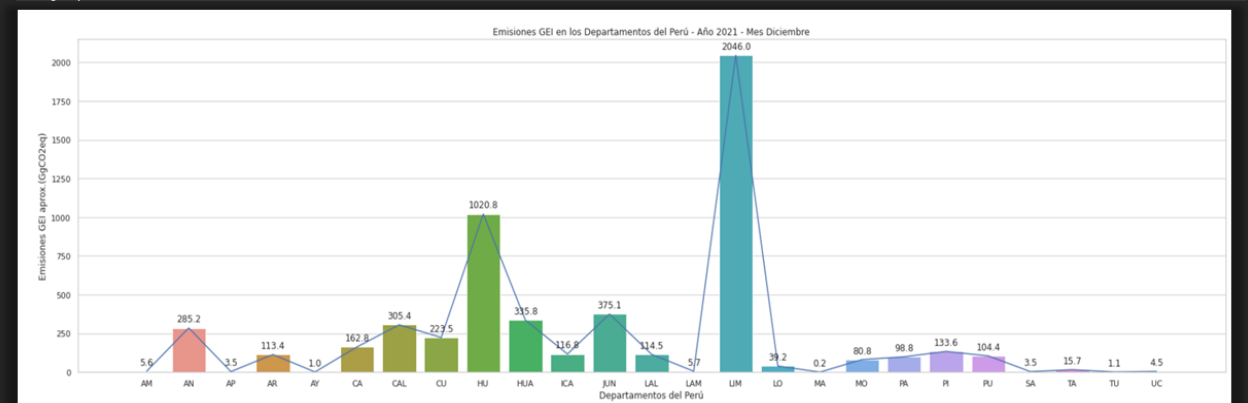
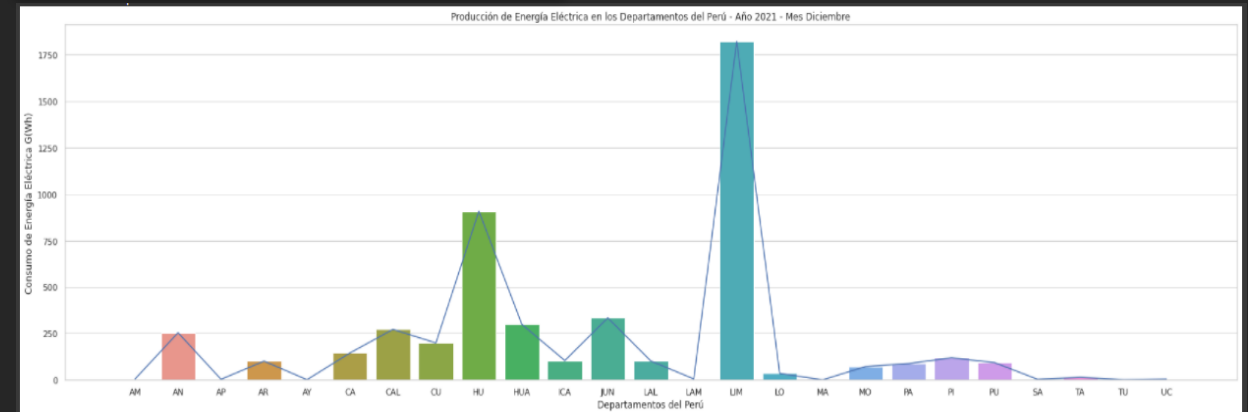
Cantidad Total de Departamentos: 25
Total Nacional de Energía Eléctrica(GWh): 4984.253000000001
Total Nacional de Emisiones GEI(GgCO₂eq): 5597.03
Valores Picos de Energía: 9

Nº	Consumo (GWh)	Departamentos	GEI aprox. (GgCO ₂ eq)
1	254	ANCASH	285.23
2	101	AREQUIPA	113.42
3	272	CALLAO	305.44
4	909	HUANCAVELICA	1020.76
5	334	JUNIN	375.06
6	1822	LIMA	2046.0
7	119	PIURA	133.63
8	14	TACNA	15.72
9	4	UCAYALI	4.49
Total:	3829		4299.75

Resultados Máximo Valor Pico: Diciembre 2021

Cantidad Total de Departamentos: 25
Total Nacional de Energía Eléctrica(GWh): 4984.253
Total Nacional de Emisiones GEI(GgCO₂eq): 5597.03
Valores Picos de Energía: 0

Nº	Consumo (GWh)	Departamentos	GEI aprox. (GgCO ₂ eq)
1	1822	LIMA	2046.0
Total:	1822		2046.0



Elaboración propia extraída de Jupyter Notebook

Discusión

- En el presente trabajo se obtuvieron los resultados donde los departamentos con altos valores picos de consumo de energía eléctrica son: Ancash con un consumo de 254 GWh, Arequipa con 101 GWh, Callao con 272 GWh, Huancavelica con 909 GWh, Junín con 334 GWh, Lima con 1822 GWh, Piura con 119 GWh, Tacna con 14 GWh y Ucayali con 4 GWh.

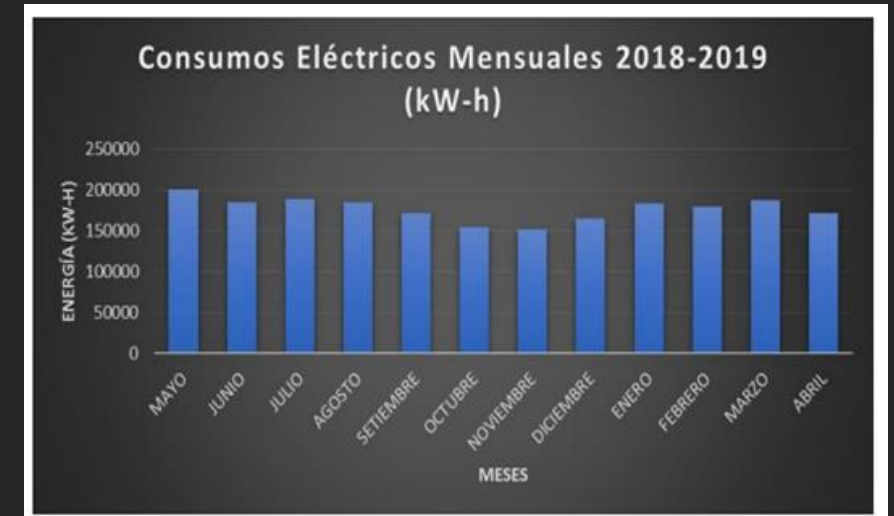


Gráfico de barras del consumo eléctrico mensual del Hospital Regional de Lambayeque, Monteza Rojas (2020)

- De forma similar Monteza Rojas (2020) de la Universidad Católica Santo Toribio de Mogrovejo, realizó un trabajo de Investigación sobre Implementar un plan de auditoría y eficiencia energética del Hospital Regional Lambayeque basado en la norma ISO 50001 para mejorar el índice de consumo energético enfatizando en la reducción de emisiones de gases contaminantes, obteniendo también los consumos eléctricos mensuales, promedio y total de dinero; dándole importancia al ahorro de energía eléctrica.

Conclusiones

- Se logró desarrollar un sistema de control y prevención usando las estrategias de programación divide y vencerás y fuerza bruta, para disminuir el consumo irresponsable de energía eléctrica y emisiones GEI de los departamentos del Perú en Jupyter Notebook Colab con el lenguaje de programación Python,
- Se analizó la complejidad de los dos algoritmos empleados, evaluándose que el de Divide y Vencerás, a una mayor cantidad de información (valores de energía de otros meses o años) usará pocos recursos, en el tiempo que lleve a ejecutarse y la cantidad de espacio de memoria que requerirá.



En la Tierra la principal fuente de energía es el sol.

Trabajos futuros y recomendaciones

- Se propone ampliar la investigación, para conocer y/o hacer pronósticos sobre las emisiones GEI, en otros años con sus meses respectivos, para un análisis de las estadísticas más profundas.
- Se propone que este sistema sea usado por autoridades, compañías energéticas y asociaciones especializadas en análisis de control y prevención de emisiones GEI que se producen debido al consumo de energía eléctrica.



La energía eólica es la energía que se obtiene a partir del viento.

Referencias

- Corporación Financiera de Desarrollo - COFIDE (2018) Inventario de Gases de Efecto Invernadero.
- GitBook (2019) Complejidad (Big-O). Notación Big-O.
- Mañas, J. A. (2017) Análisis de Algoritmos – Complejidad. Dept. de Ingeniería de Sistemas Telemáticos.
- MEM (2022). Ministerio de Energía y Minas Perú. Principales Indicadores Del Sector Eléctrico A Nivel Nacional enero 2022 (Cifras Preliminares Al Mes De diciembre 2021). Producción de Energía Eléctrica Nacional. Recuperado de: <http://www.minem.gob.pe/minem/archivos/2%20Cifras%20preliminares%20del%20Sector%20Elctrico%20-%20Diciembre%202021a.pdf>
- MINAM (2016). Ministerio del Ambiente Perú. Inventario Nacional De Gases De Efecto Invernadero 2016. Emisiones de GEI. Recuperado de: <https://infocarbono.minam.gob.pe/annios-inventarios-nacionales-gei/ingei-2016/>
- Ministerio de Energía y Minas (2020) Uso eficiente de la Energía Guía Metodológica para docentes de Secundaria.
- Monteza Rojas, L. E. (2020). Implementar un plan de auditoría y eficiencia energética del Hospital Regional Lambayeque, basado en la norma ISO 50001 para reducir los consumos energéticos.
- Organismo de Evaluación y Fiscalización Ambiental - OEFA (2015). La supervisión ambiental en el subsector electricidad.
- Osinergmin (2016). Organismo Supervisor de la Inversión en Energía y Minería. Reporte Semestral De Monitoreo Del Mercado Eléctrico Segundo Semestre De 2016. Producción Total De Energía Eléctrica A Nivel Nacional.
- Tamayo, Jesús; Salvador, Julio; Vásquez, Arturo y Carlo Vilches (Editores) (2016). La industria de la electricidad en el Perú: 25 años de aportes al crecimiento económico del país. Osinergmin. Lima, Perú.



¡Gracias!