[[1]](#footnote-0)

RETO 1: CÁLCULO DE LA CAPACIDAD DE UN SISTEMA DE ENERGÍA SOLAR

Autores: González Gonzáles Dinalut y Ruiz Rincon Jessica Jineth estudiantes de la fundación universitaria de San gil unisangil (SEDE-CHIQUINQUIRA)

***Resumen*—****En el presente documento hemos diseñado un algoritmo en Python para calcular cuántos paneles solares necesita una casa para cubrir su consumo energético anual. Principalmente aplicamos una metodología donde establecimos los requisitos y limitaciones del algoritmo. Además utilizamos PSeInt para crear un pseudocódigo que luego implementamos en Python. Al ejecutar el programa logramos identificar que el algoritmo calcula de forma precisa**

**Palabras clave-cálculo, capacidad, sistema, energía, solar, Python. Programación**

INTRODUCCIÓN

 En este documento se busca diseñar un algoritmo que permita calcular cuántos paneles solares necesita una casa para cubrir su consumo energético anual. Para ello es necesario aplicar una metodología que identifique y calcule los datos necesarios  para la implementación de un lenguaje de programación  como lo es Python.

1. *ANÁLISIS*
2. *Contexto*

Una familia quiere instalar un sistema de energía solar para cubrir su consumo de electricidad anual y ser más sostenible.

*B. Población*

Familia que quiere instalar un sistema de energía solar para cubrir su consumo de electricidad anual y ser más sostenible.

*C. Limitaciones y alcance*

Las limitaciones y alcances que se pudieron identificar son las siguientes:

**Limitaciones**

* Datos necesarios para el cálculo de la capacidad del sistema de energía solar.
* Exactitud en los datos ingresados
* Área que ocupa el panel solar
* Radiación solar
* Eficiencia del panel

**Alcances**

* El usuario podrá ingresar los datos requeridos
* Se calculará la potencia diaria de un panel solar
* Se calculará la potencia anual de un panel solar
* Se calculará cuántos paneles solares son necesarios para cubrir el consumo anual de 12.000 kwh
* Se calculara el área total para la instalación de paneles

III. OBJETIVOS

1. *Objetivo general*

Diseñar un algoritmo donde el usuario pueda ingresar los datos de consumo energético y los del panel solar. Para saber cuántos paneles debe instalar para cubrir su consumo

1. *Objetivos específicos*

* Solicitar al usuario los datos de consumo y del panel solar.
* Determinar el lenguaje de programación que se va a usar para crear el algoritmo.
* Generar la salida correspondiente de los datos.
* Presentar al usuario un programa que pueda manejar de forma comprensible.

IV. ESPECIFICACIÓN DE REQUERIMIENTOS.

1. *CON RESPECTO AL DISPOSITIVO*

A continuación se indican los requerimientos mínimos:

* Mínimo 4 GB de RAM.
* Sistema operativo: Windows 8 en adelante.
* CPU: Intel Core i3.
* Disco: HDD o SSD.

1. *CON RESPECTO AL USUARIO.*

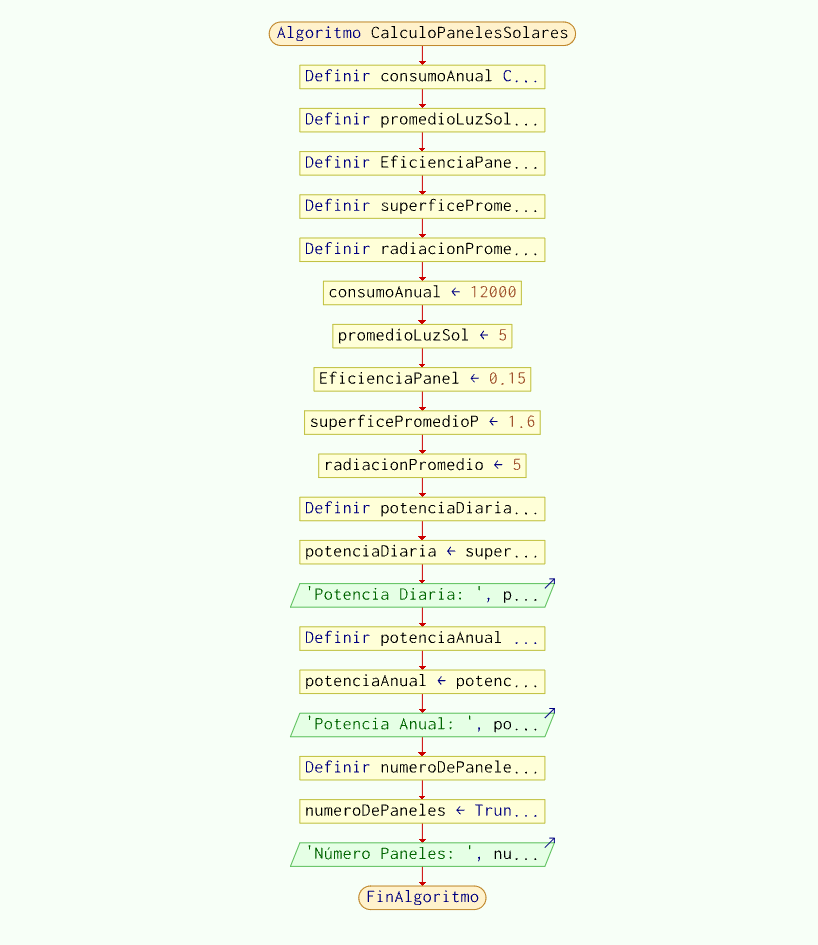
*Requisitos mínimos*

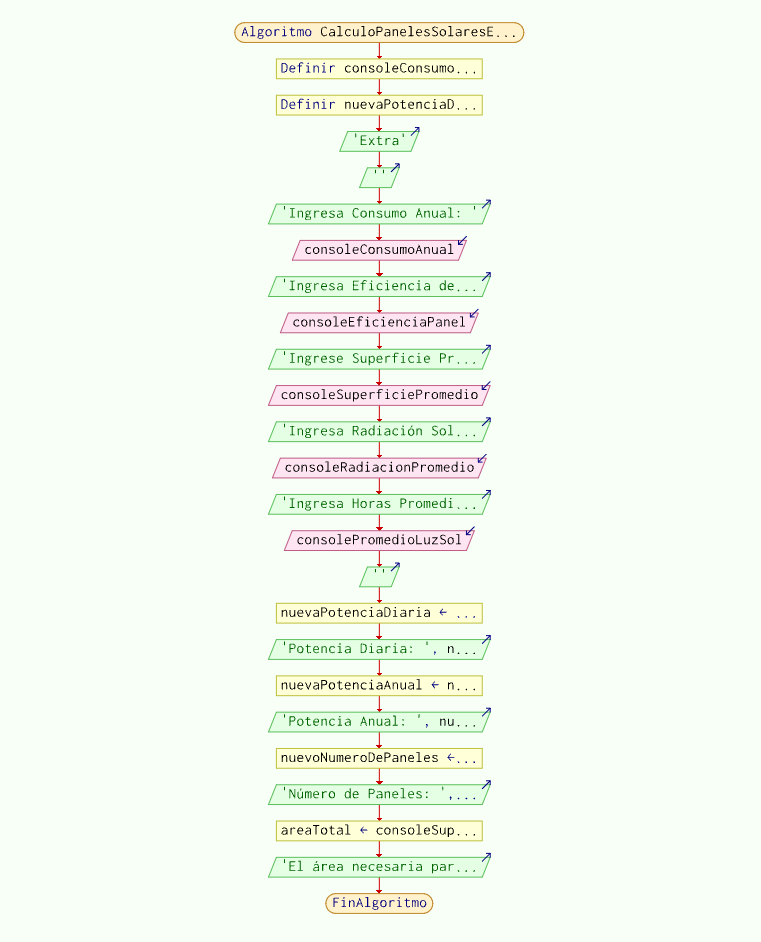
* El usuario debe ingresar datos correctos.
* El usuario deberá tener un área disponible para insertar paneles solares.

V. DISEÑO DEL ALGORITMO

Utilizamos un programa generador de pseudocódigo denominado PSeInt, el cual utilizamos  como guía, para luego usar el lenguaje de programación Python, siendo este el estructurador del software requerido.

Figura. 1. Pseudocódigo.





En esta imagen. Se muestra el prototipo principal a tener en cuenta para el diseño del código fuente

VI.CODIFICACIÓN.

 Para la codificación del algoritmo utilizamos los siguientes  programas:

* PSeInt.
* Visual Studio Code.
* python
* Git hub.

*Instrucciones del código fuente*

1. Se definen variables con un contexto Global las cuales son: consumoAnual, promedioLuzSol, eficienciaPanel, superficiePromedioP, radiacionPromedio.
2. La variable potenciaDiaria es definida por la operación: superficePromedioP \* radiacionPromedio \* eficienciaPanel.
3. Un print posterior será el encargado de imprimir el resultado de la variable potenciaDiaria.
4. La variable potenciaAnual es definida por la operación: potenciaDiaria \* 365.
5. Un print posterior será el encargado de imprimir el resultado de la variable potenciaAnual.
6. La variable numeroDePaneles es definida por la operación: consumoAnual / potenciaAnual.
7. Un print posterior será el encargado de imprimir el resultado de la variable numeroDePaneles.
8. Se genera otro requisito en el pseudocódigo el cual es recolectar los valores que el usuario digite por consola, los cuales son: consoleConsumoAnual, consoleEficienciaPanel, consoleSuperficiePromedio, consoleRadiacionPromedio, consolePromedioLuzSol
9. La variable nuevaPotenciaDiaria es definida por la operación: consoleSuperficiePromedio \* consoleRadiacionPromedio \* (consoleEficienciaPanel / 100)
10. Un print posterior será el encargado de imprimir el resultado de la variable nuevaPotenciaDiaria.
11. La variable nuevaPotenciaAnual es definida por la operación: nuevaPotenciaDiaria \* 365.
12. Un print posterior será el encargado de imprimir el resultado de la variable nuevaPotenciaAnual.
13. La variable nuevoNumeroDePaneles es definida por la operación: consoleConsumoAnual / nuevaPotenciaAnual.
14. Un print posterior será el encargado de imprimir el resultado de la variable nuevoNumeroDePaneles.
15. La variable areaTotal es definida por la operación: consoleSuperficiePromedio \* nuevoNumeroDePanles. La cual servirá para hallar el área necesaria para la instalación de paneles.
16. Un print posterior será el encargado de imprimir el resultado de la variable nuevoNumeroDePaneles y también la variable areaTotal.

*B. Código fuente*

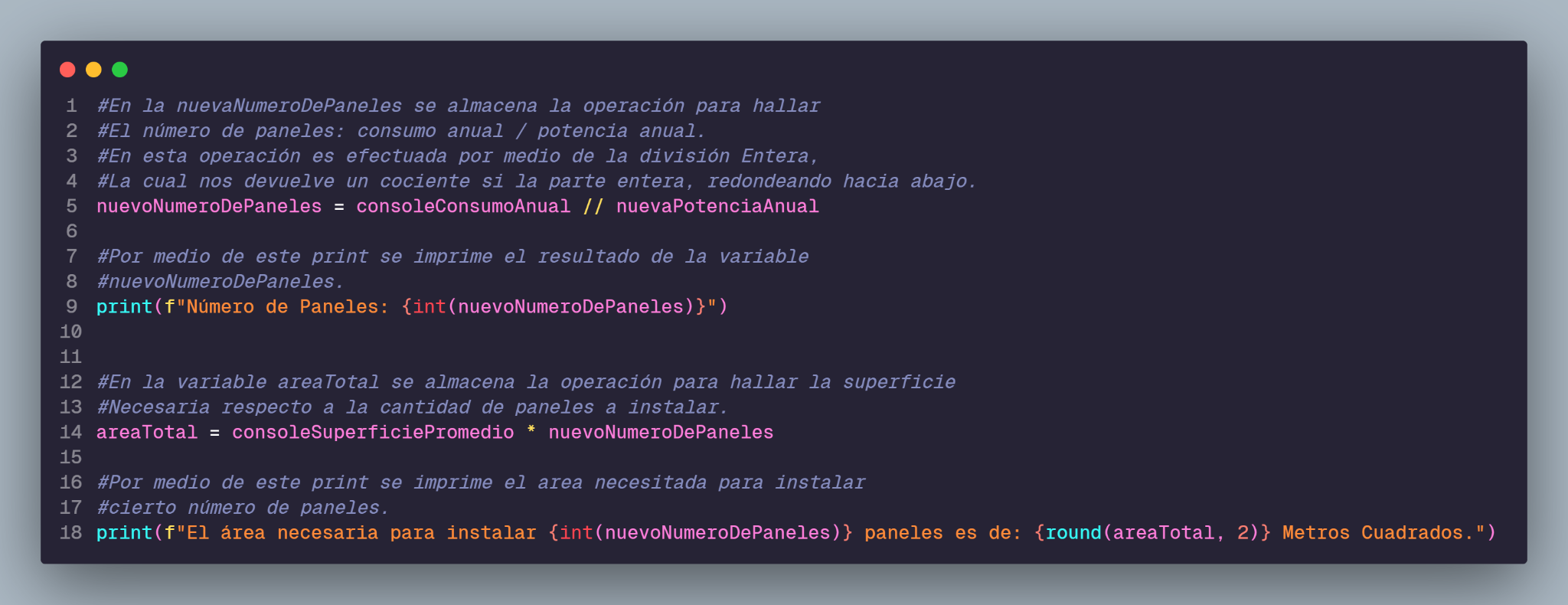
Imagen. 1. Código fuente.



Imagen. 2. Código fuente.



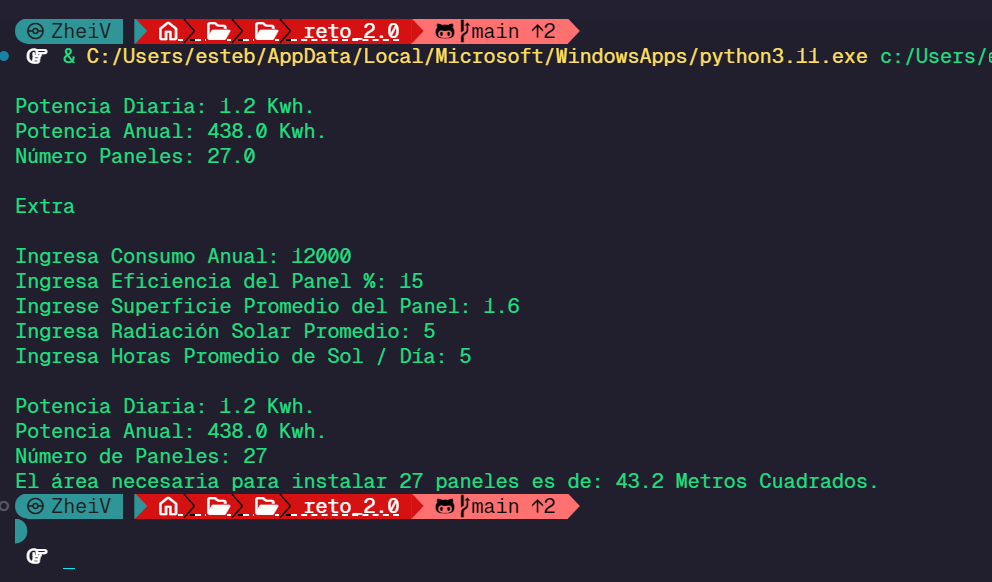
Imagen. 3. Código fuente.



*VII. EJECUCIÓN DEL PROGRAMA.*

En este caso comenzamos con la apertura de la terminal de Git Bash. Si ya se creó el repositorio remoto y se sincronizan las cuentas de los desarrolladores con el repositorio local, se envía el archivo al Staging area con el comando git add.  Luego ejecutamos el git commit -m “Comentario” para confirmar los cambios, donde, posteriormente usamos git push origin o pb master o main dependiendo la rama que se esté usando con el fin de almacenar la información en el repositorio remoto de GitHub.

Imagen. 3. Código ejecutado.



Nota: en esta imagen se muestra que efectivamente el algoritmo funciona cuando el usuario digita los datos requeridos

VIII CONCLUSIONES.

* En el presente trabajo logramos describir el  diseño y codificación de un algoritmo el cual permite al usuario calcular el consumo anual generado por sus paneles solares.
* Implementamos un análisis situacional específico para captar con mayor detalle los requisitos del cliente, los cuales gracias a dicho análisis pudimos utilizar para escribir el código de manera correcta
* Incorporamos la evaluación y pruebas para el programa, con ellas constatamos y verificamos que el código realizado por nosotros los desarrolladores esté a fin con los requerimientos del cliente.
* Aprendimos que es necesario aplicar una metodología que hace más comprensible y efectivo el proceso a la hora de desarrollar un algoritmo.
* Empleamos un generador de pseudocódigo denominado PSeInt como guía para estructurar de una forma clara el algoritmo en Python.
* Para realizar la codificación del código utilizamos varios programas como Visual Studio Code y GitHub los cuales facilitan el trabajo en grupo y la gestión de información.

1. Documento elaborado el 15 de septiembre 2024 [↑](#footnote-ref-0)