**CÁLCULO DE PORCENTAJE DE RETRASOS EN RELACIÓN CON EL TIEMPO ASIGNADO A UNA TAREA(Septiembre 2024)**

Autores: Torres Pachón Juan Esteban, González Gonzalez Dinalut y Ruiz Rincon Jessica Jineth estudiantes de la foundation universitaria de San gil unisangil (SEDE-CHIQUINQUIRA)

***Resumen*****—En el presente documento hemos diseñado un algoritmo en python para calcular el retraso en días de las tareas completadas fuera de plazo y el porcentaje de retraso en relación con el tiempo total asignado a la tarea. 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, Días, Retrasos, Python. Programación**

1. INTRODUCCIÓN

 En este documento se busca diseñar un algoritmo que permita calcular el retraso en días de las tareas completadas fuera de plazo y el porcentaje de retraso en relación con el tiempo total asignado a la tarea. Para ello es necesario aplicar una metodología que identifique y calcule los datos necesarios  para la implementación de un lenguaje de programación  como Python.

1. *ANÁLISIS*

*A. Contexto*

Desarrollar un algoritmo para una empresa de desarrollo de software que gestione varios proyectos simultáneamente. Cada miembro del equipo tiene tareas asignadas con fechas límite específicas.

*B. Población*

Empresa de desarrollo de software que gestiona varios proyectos simultáneamente.

*C. Limitaciones y alcance*

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

*Limitaciones*

* Datos necesarios para calcular el retraso en días de las tareas completadas fuera de plazo y el porcentaje de retraso

*Alcance.*

* El usuario podrá ingresar los datos requeridos
* Se calculará los días de retraso
* Se calculará el porcentaje de retrasos

III. OBJETIVOS

*Objetivo general*

Diseñar un algoritmo donde el usuario pueda ingresar los datos, fecha de entrega y día en que entrego. Para saber cuántos días se retrasó la entrega y el porcentaje de retrasos

1. *Objetivos específicos*

* Solicitar al usuario la fecha de entrega, y la fecha en que entrega.
* 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. C*ON 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.*

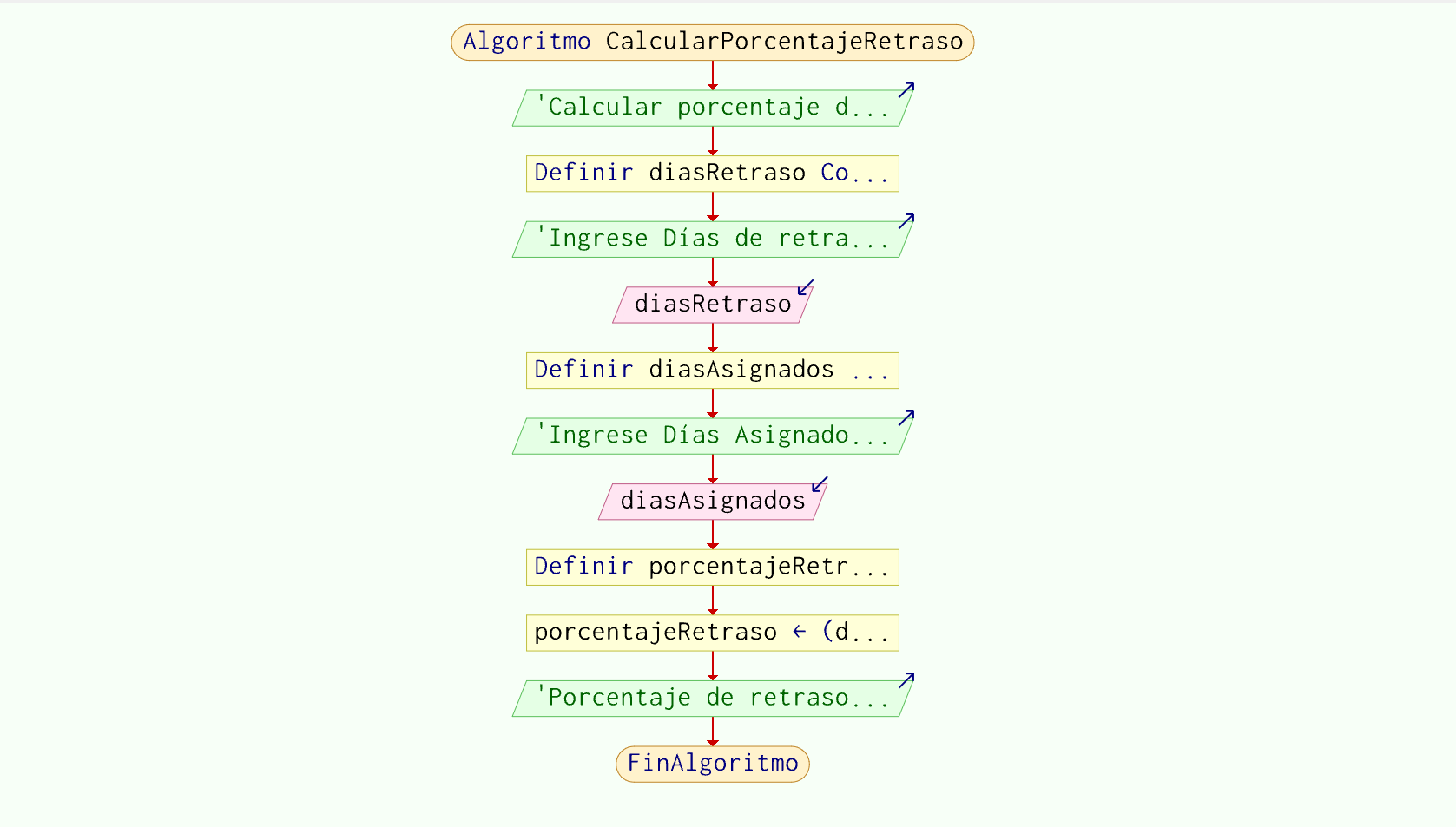
*Requisitos mínimos*

* *El usuario debe ingresar datos correctos.*
* *Los datos a ingresar deben ser números entero para que el usuario obtenga una respuesta precisa.*

V. DISEÑO DEL ALGORITMO

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

Imagen. 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. Solicitar al usuario fecha en que entregó.
2. Solicitar al usuario fecha de entrega
3. Calcular los días de retraso.
4. Calcular el porcentaje de retrasos
5. Imprimir porcentaje de retrasos

*B. Código fuente*

Figura. 2. Código fuente.

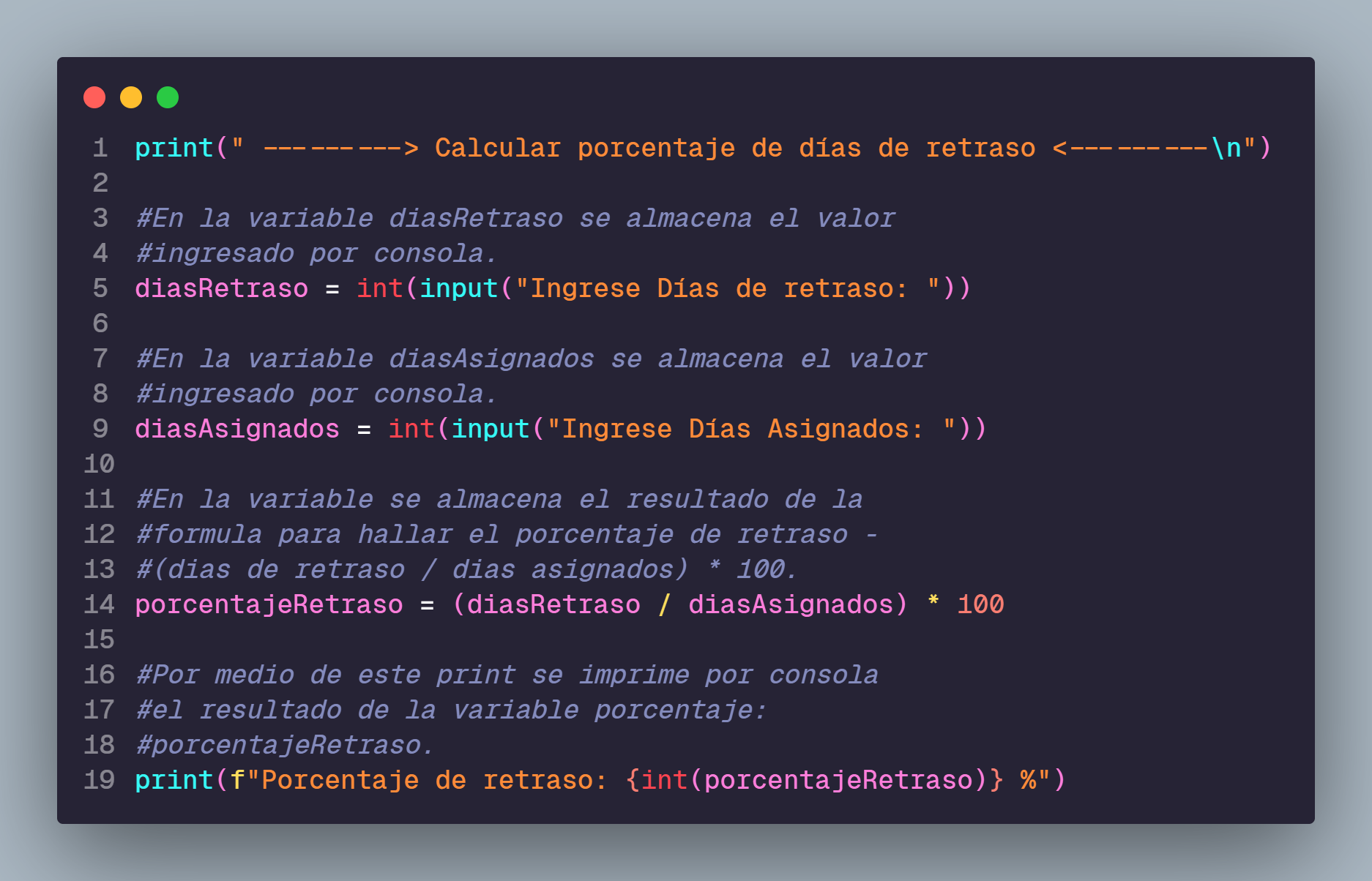
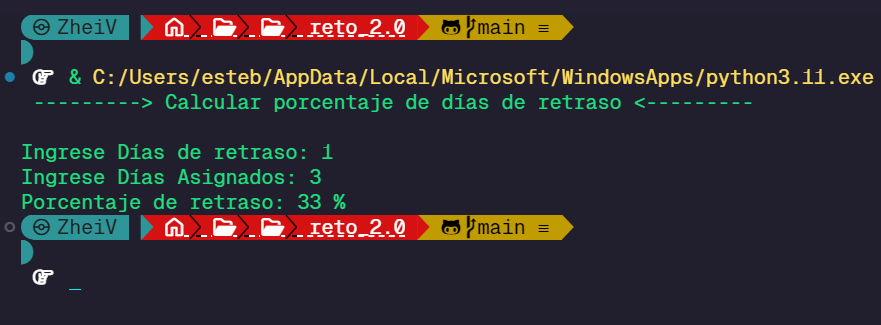
****

Imagen. 2: En esta imagen se muestra el diseño del algoritmo donde se utiliza el lenguaje de programación python para ejecutarlo.

*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 porcentaje de días que se retrasó la entrega.
* Utilizamos un enfoque de investigación cuantitativo el cual nos permitió rápidamente determinar la salida correcta de la aplicación según los requerimientos propuestos por el cliente.
* 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.