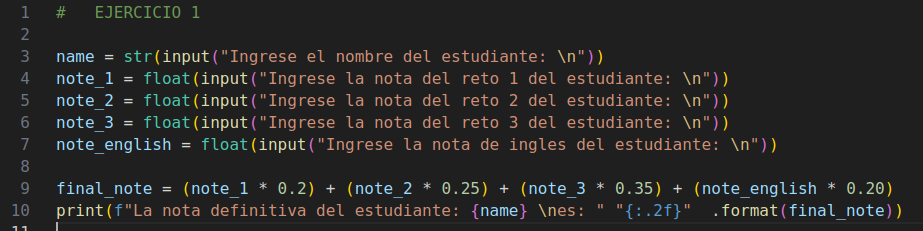
**Taller**

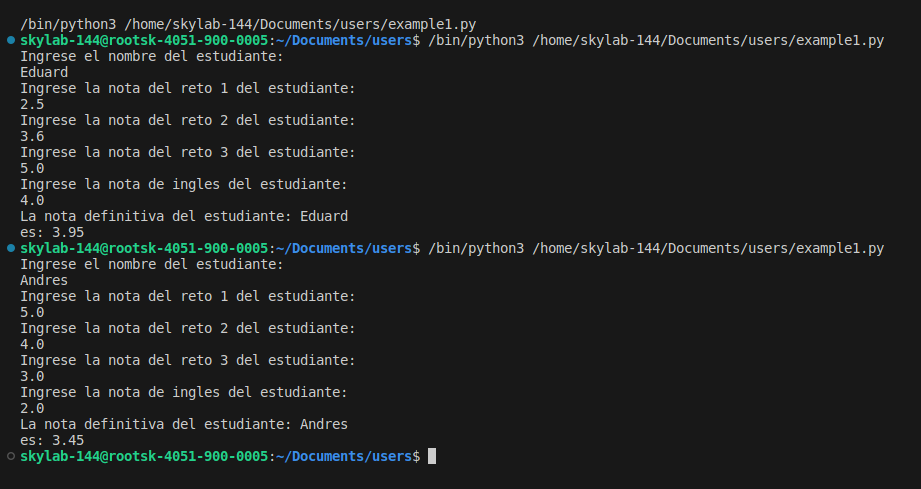
Aplique el ciclo de vida de los algoritmos dando solución a los siguientes enunciados.

1. Sobre un estudiante conocemos las calificaciones parciales obtenidas en los retos, nota reto 1, nota reto 2, nota reto 3 y nota de inglés y el nombre del estudiante. Si los porcentajes para cada reto de inglés son 20%, 25%, 35% y 20% respectivamente, calcular su calificación definitiva e imprimirla, además del nombre

* Análisis: El cliente requiere un programa para calcular las notas definitivas de un estudiante según el porcentaje o valor que tienen y sumarlas, no requiere ciclos y por cada ejecución puede hacer el cálculo de las notas de 1 estudiante.
* Diseño: Se necesitan 5 variables para el nombre, las 3 notas y la nota de inglés, las interacciones se pueden guardar en una variable de la nota final y por último se hace una impresión de los datos.
* Codificación: Se escribe el código anteriormente diseñado y obtenemos este resultado:



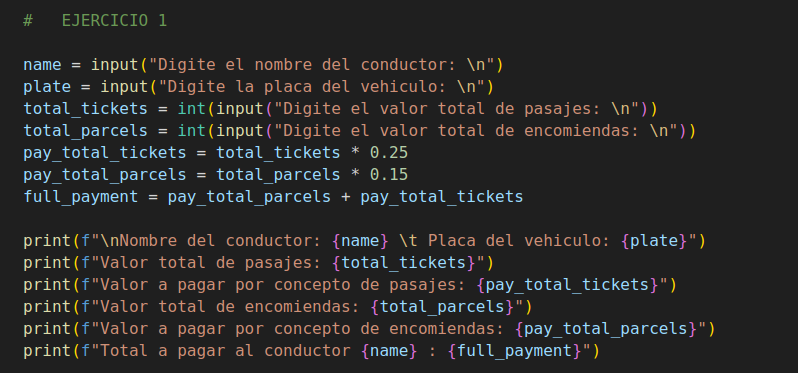
* Pruebas: Realizamos varias pruebas observando su óptimo funcionamiento.



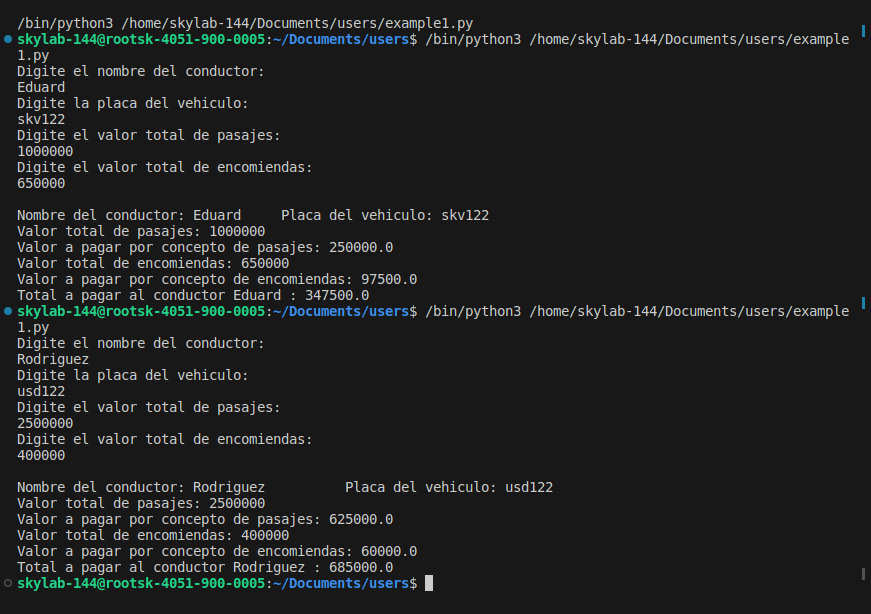
* Implementación: El programa está listo para su uso.
* Mantenimiento: Se evidencia que el programa no necesita de próximo mantenimiento puesto cumple con los requerimientos del enunciado.

2. Se desea liquidar el valor a pagar a un conductor de una buseta de transporte intermunicipal. Se conoce el nombre, placa del vehículo, valor total por concepto de pasajes y el valor total por concepto de encomiendas. Si por el valor de los pasajes se le liquida el 25% y por el valor de encomiendas se le liquida el 15%, se pide calcular el valor total a pagar al conductor. Se debe visualizar, el nombre, placa del vehículo, valor total de pasajes, valor a pagar por concepto de pasaje, valor total de encomiendas, valor a pagar por concepto de encomiendas y el valor total a pagar al conductor.

* Análisis: La solicitud consta de un código para una empresa de transporte intermunicipal el cual necesita calcular el valor para pagar a los conductores dependiendo de los conceptos mencionados: pasajes y encomiendas.
* Diseño: Las variables con datos de entrada son el nombre del conductor, la placa, el total de pasajes y el total de las encomiendas, necesitamos 3 variables, 2 para el cálculo de los conceptos y uno para el total del pago del conductor, finalmente se muestra toda la información clara y detalladamente
* Codificación: Se escribe el código anteriormente diseñado y obtenemos este resultado:



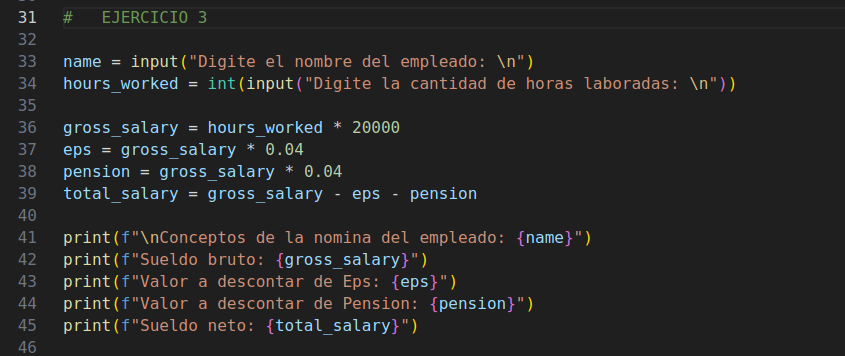
* Pruebas: Realizamos varias pruebas observando su óptimo funcionamiento.



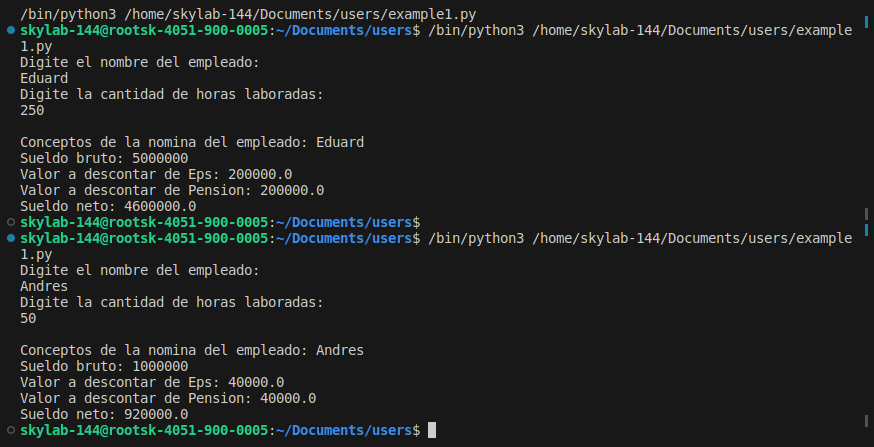
* Implementación: El programa está listo para su uso.
* Mantenimiento: Se evidencia que el programa no necesita de próximo mantenimiento puesto cumple con los requerimientos del enunciado.

3. La empresa ACME desea calcular el valor de la nómina de un empleado, tanto el sueldo bruto como el sueldo neto. El sueldo bruto se calcula a partir del valor de la hora y la cantidad de horas trabajadas. A esto se le descuenta el valor de la EPS que es el 4%, el valor de la Pensión que es el 4%. El sueldo neto es el sueldo bruto menos los descuentos. Mostrar al final, el valor del sueldo bruto, cada uno de los descuentos y el valor del sueldo neto. Para este ejercicio el valor de la hora es $20.000.

* Análisis: Se requiere un programa para la empresa ACME en el que se quiere calcular el sueldo de un empleado y hacer los respectivos descuentos de la EPS y de la Pensión teniendo en cuenta un porcentaje estable, se especifica además el valor de la hora del trabajador.
* Diseño: Se recibe el dato del nombre del empleado y las horas trabajadas, el cálculo para las horas se hacen teniendo en cuenta el valor de la hora especificado en el enunciado, luego en variables respectivas para la EPS y Pensión se hacen sus cálculos y por último se imprimen los valores obtenidos.
* Codificación: Se escribe el código anteriormente diseñado y obtenemos este resultado:



* Pruebas: Realizamos varias pruebas observando su óptimo funcionamiento.



* Implementación: El programa está listo para su uso.
* Mantenimiento: Se evidencia que el programa no necesita de próximo mantenimiento puesto cumple con los requerimientos del enunciado.

4. Hacer un algoritmo que, dada una hora con minutos y minutos adicionales, calcule la nueva hora. Asuma que no se pasa de las 23:59

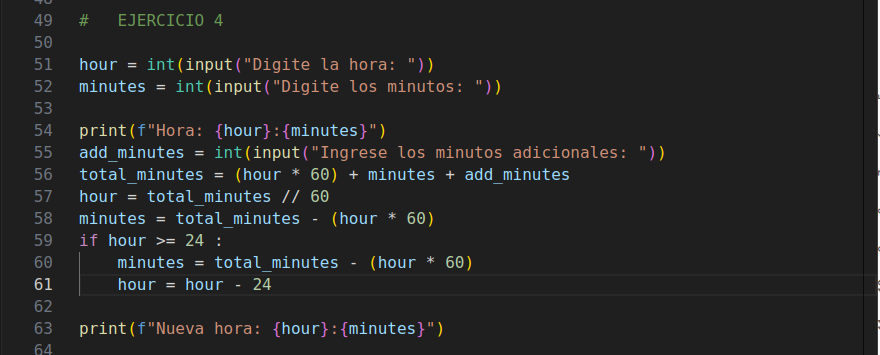
Ejemplo

Hora: 10:45

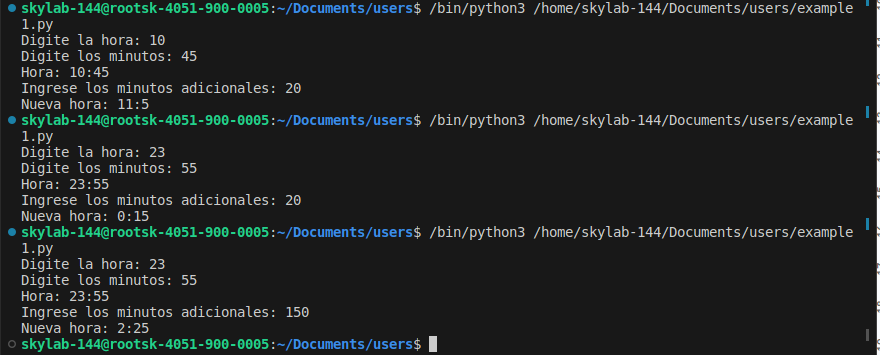
Minutos: 20

Nueva Hora: 11:05

* Análisis: El enunciado necesita que al ingresar una hora en específico se pueda sumar otro valor adicional de minutos y también hay una condición que la hora no puede pasar de las 23:59.
* Diseño: Se reciben 2 datos inicialmente, tanto el de la hora como el de los minutos, por lo tanto esos datos se van a poder convertir en minutos para trabajar en el programa con un concepto de la variable para evitar confusiones, luego se crea la condicional para validar si el valor de la hora ha superado las 24 horas, en ese caso se restan las 24 horas para que inicie desde cero el contador de las horas, en caso de que sean más de 24 horas nos indica la 1 o 2 dependiendo de los minutos adicionales.
* Codificación: Se escribe el código anteriormente diseñado y obtenemos este resultado:



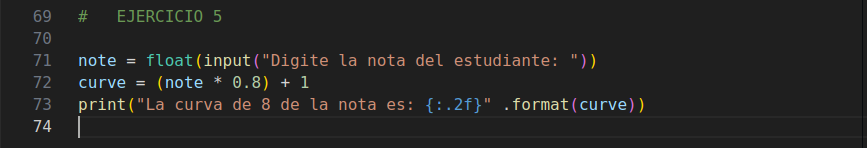
* Pruebas: Realizamos varias pruebas observando su óptimo funcionamiento.



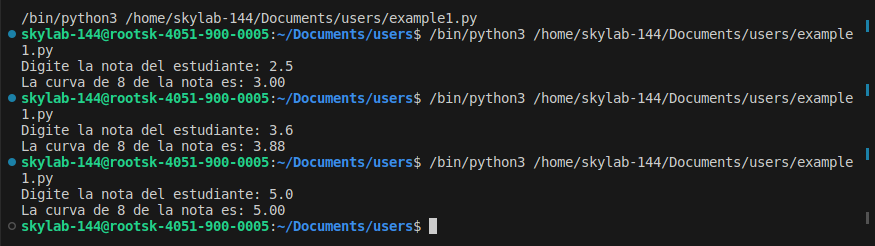
* Implementación: El programa está listo para su uso.
* Mantenimiento: Se evidencia que el programa no necesita de próximo mantenimiento puesto cumple con los requerimientos del enunciado.

5. Hacer un algoritmo que, dada una nota (de 0.0 a 5.0), calcule la curva de 8 de la nueva nota. La curva de 8 se calcula multiplicando la nota por 0.8 y sumándole 1.

* Análisis: El enunciado solicita que podamos ingresar una nota como valor y nos especifica la forma en la cual se debe calcular la denominada “Curva de 8”
* Diseño: Se recibe el dato de la nota y se hace el cálculo de la curva de 8 teniendo en cuenta las indicaciones, el cual se multiplica la nota por 0.8 y se suma 1.
* Codificación: Se escribe el código anteriormente diseñado y obtenemos este resultado:



* Pruebas: Realizamos varias pruebas observando su óptimo funcionamiento.



* Implementación: El programa está listo para su uso.
* Mantenimiento: Se evidencia que el programa no necesita de próximo mantenimiento puesto cumple con los requerimientos del enunciado.