

**Universidad Técnica Particular de Loja**

Área Técnica

Informática

Ingeniería de Software

**Tema**

**Tarea del Segundo Bimestre:**

Autor:

**Juan Morocho**

Profesor:

**Abad Espinoza Marco Patricio**

16/01/2019

# Actividad 2:

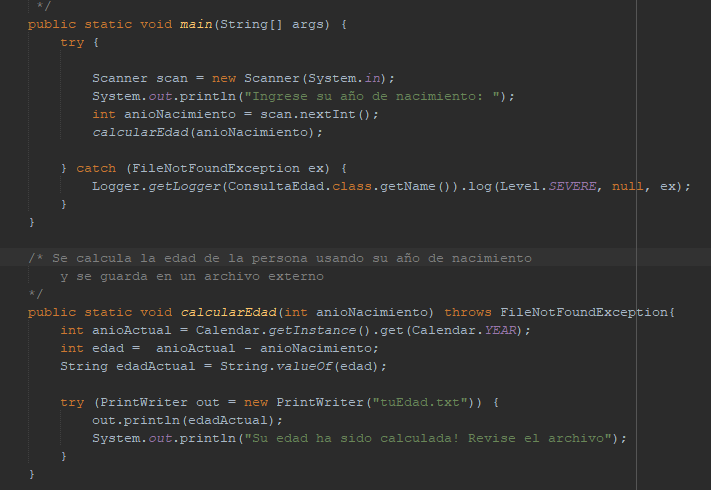


Ilustración 1: Código Fuente

Se tiene una función calcular edad que recibe como parámetro el año de nacimiento, con esta variable nos ayudara a calcular la edad de la persona se obtiene el año actual y se procede a restar el año actual menos el año de nacimiento y se obtiene la edad se convierte le edad de tipo entero a una cadena para imprimir dicha cadena en el archivo “tuEdad.txt” ubicado dentro de la carpeta ConsultaEdad.

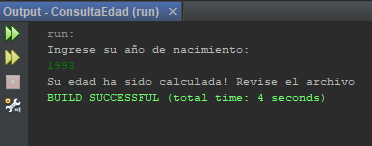


Ilustración 2: Corrida del programa

Cuando de ejecuta el programa se pide al usuario ingresar su año de nacimiento una vez ingresado se le dirige al usuario revisar el directorio del proyecto donde encontrara un archivo con edad.

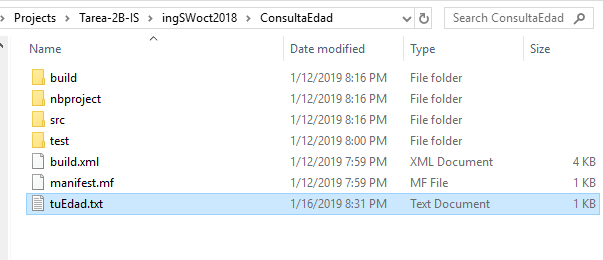


Ilustración 3: Archivo con la edad calculada

Dentro del directorio del proyecto se ve el archivo “tuEdad.txt” que contiene el resultado final de la edad del usuario.

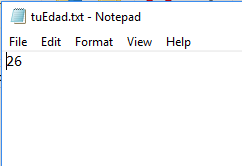


Ilustración 4: La edad del usuario

**Comentario Personal:**

Para el trabajo de un desarrollador, es fundamental utilizar sistemas de control de versiones para todos los archivos para permitir que los desarrolladores o equipos múltiples trabajen de manera aislada sin afectar el trabajo de otros. Este aislamiento permite que las características se construyan, prueben, integren o incluso desechen de una manera controlable, transparente y mantenible.

**Funcionalidad básica**

Los conceptos básicos del control de versiones son la capacidad de guardar los cambios realizados en los archivos, mientras se conservan los cambios de todas las versiones anteriores. Todos los cambios realizados dentro de Git reciben un identificador de versión único junto con un comentario escrito por el usuario donde se pueden describir los cambios.

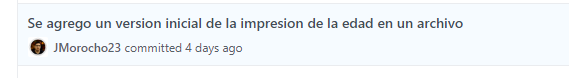


Ilustración 5: Ejemplo de un comentario en el commit

**Commits**

Los cambios en los archivos son monitoreados a nivel de usuario. Cuando se realizan cambios en un espacio de trabajo local de los desarrolladores, hacen commit (guardan localmente) y, a continuación, envían al repositorio dentro de Git, esto lo hace disponibles para el consumo de los otros desarrolladores del equipo. Esos desarrolladores hacen pull para recuperar nuevas actualizaciones del repositorio todos los cambios en su propio espacio de trabajo local. Este flujo permite a los desarrolladores trabajar en una variedad de características dentro de un proyecto sin afectar el trabajo de otros. Una vez que han completado su función, se envía al repositorio y se pone a disposición del resto de equipo.

**Branches**

Los branches dentro de Git es una forma de llevar una copia del código a lo que se llama un branch. Esto permite a los desarrolladores aislar todos sus cambios dentro de esa rama antes de fusionarlos aplicando los cambios a la copia principal del código una vez que se complete la función en la que se ha trabajado. Varios desarrolladores pueden trabajar en una rama simultáneamente.

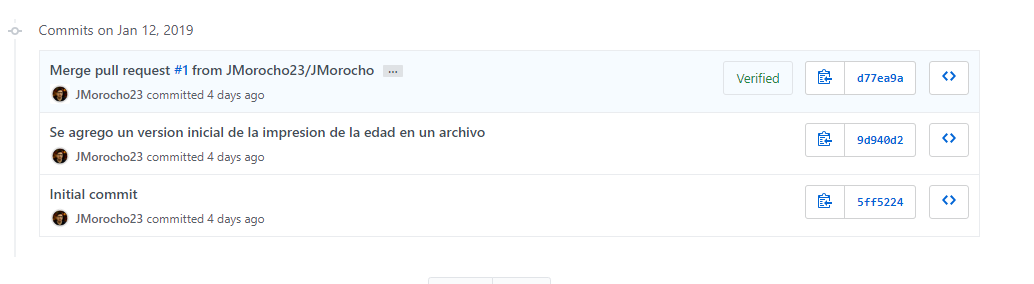


Ilustración 6: Ejemplo de commits con branches