**La Fábula del Pastor y el Líder de Proyectos**

Me pareció interesante la fábula, nos explica a que todo líder debe de entender y comprender lo que esta pasando en su entorno, y que para tener las mejores ganancias debe de crear los mejores entornos para que sea posible él trabajo, no solo hacer que la gente trabaje, nos lo explica de manera sencilla y ponen como ejemplo a un pastor de ovejas de no mas de 50 recursos y un líder de proyectos, en donde el líder de proyectos queda fascinado al ver que el pastor tenia un excelente control sobre las ovejas, y el líder tenia dificultades con sus trabajadores.

El pastor vio al líder y le convido de su licor que tenia y a lo cual el líder se emborracho, y entraron en un momento de discusión a tal punto de cambiar de roles, el líder de proyectos fracaso en cuento a hacer la labor que hacia el pastor, ya que no pudo completar su plan que tenía, el cual era entregar cierta cantidad de lana al alcalde, el cual no pudo completar, en el proceso nunca les hizo caso a las ovejas, ellas le comentaban mejoras que podría hacer pero el líder de proyectos nunca es hizo caso, solo quería que siguieran produciendo.

Por otro lado, el pastor que tuvo que ir a la empresa del líder de proyectos, pero sin embargo todo fue un éxito, ya que les hizo caso a los trabajadores, si necesitaban algo el pastor lo conseguía con el fin de que fueran más eficientes, el líder no lograba entender como el pastor había logrado, el pastor le conto y el líder quedo impactado, pero cuando se dio cuenta despertó por que era un sueño a lo lejos vio al pastor irse con sus ovejas.

**JDK de Java**

A lo que leí y entendí es JDK de Java es un kit de desarrollo de java, el cual es necesario para poder compilar, depurar programas hechos en java, al decir que es un kit entiendo que contiene múltiples cosas (máquina virtual, bibliotecas) las cuales se complementan.

Nota indagando encontré esto:

**Compilador javac**: convierte archivos .java (código fuente) en archivos .class (bytecode ejecutable por la JVM).

Me día la tarea a hacer mi primer programa en java con el editor de código visual studio code, a lo cual instalé una extensión, para ejecutar el programa, a lo cual no tuve problemas.

Posteriormente indagando y leyendo observe que también lo podía ejecutar desde la terminal de mi computadora, a lo cual tuve un problema.

1.- primero observe que debía ejecutar el comando “javac HolaMundo.java”, esto para convertir mi archivo .java en un archivo .class a lo que no entendí por qué, investigue y el .class es un archivo bytecode que entiende la máquina virtual de Java.

2.- posteriormente tuve que ejecutar el comando “java HolaMundo” esto para ejecutar el programa en la maquina virtual y muestre el resultado.

A lo cual tuve una pregunta, ¿Por qué es necesario hacer esa conversión de archivo en la terminal de mi computadora y no en el editor de código visual studio code?

**Maven**

Por lo que investigue y entendí Maven es un software para la gestión de proyectos de java, el cual nos permite gestionar librerías que vamos a ocupar en los proyectos, nos ayudara a descargarlas en su última versión sin la necesidad de buscarla manualmente. utiliza un archivo POM.xml (especifica las librerías que necesitamos y Maven las descarga).

**Programación Orientada a Objetos y principios SOLID**

En el curso que realice nos comenta que la programación orientada a objetos es un paradigma (forma, estilo de programar), en el que hace uso de objetos para realizar o construir un software.

En la programación orientada a objetos utilizaremos clases y objetos, pero también tenemos principios en la programación orientada a objetos (POO).

* Abstracción
* Encapsulamiento
* Polimorfismo
* Herencia

(La verdad no me quedo muy claro esa parte, pero poniéndolo en práctica me ire familiarizando)

**SOLID**

Son recomendaciones que se componen de 5 principios de diseño de software, todo esto para que sea mas escalable y comprensible.

S – Single Responsibility Principle (SRP)

O – Open/Closed Principle (OCP)

L – Liskov Substitution Principle (LSP)

I – Interface Segregation Principle (ISP)

D – Dependency Inversion Principle (DIP)

(De igual manera aún no me quedo claro al cien por ciento pero conforme vaya trabajando me iré familiarizando)