

Luego de varios años impartiendo docencia he compartido con personas que han tenido cierta dificultad en aprender a programar, varios de ellos han pedido hablar conmigo en privado y me piden consejos para mejorar su rendimiento. Esta situación me motiva a escribir una serie de consejos breves y concisos que entiendo pueden ser de ayuda.

Estos consejos están escritos orientado a los alumnos que están estudiando programación bajo la plataforma **JAVA**. No obstante la esencia de las recomendaciones es genérica y entiendo que valdría la pena que los leyeras aún estés aprendiendo a programar bajo otra tecnología.

- No intente correr sin antes gatear!: No ha sido ni uno ni dos las personas que he visto que quieren aprender a programar pero están impacientes por hacer cosas productivas, y quieren "pasar a la parte interesante" y "saltar la parte aburrida". Resulta que para ser productivo y saber lo que se está haciendo la parte "aburrida" es esencial. Para muchos es desesperante durar 5 meses estudiando java y no haber hecho un solo programa de interfaz gráfica. Deben de agotar etapas, deben de aprender toda la base y LUEGO entonces pasar a temas más complejos.

Varias personas también he visto que intentan hacer programas gráficos de cierta complejidad usando eclipse o NetBeans arrastrando botoncitos con la esperanza de poder hacer programas sin saber programar, y en no más de una hora se enteran de que simplemente hay que saber programar para poder desarrollar una aplicación en Java.

En este punto también he visto personas intentando correr y entender aplicaciones de Java Enterprise Edition (JEE) sin tener si quiera los conocimientos básicos de programación Java.

- LEA! : Se van a dar cuenta que las personas que han sido exitosas programando al menos tienen un mínimo hábito de lectura. Encuentre placer en la lectura. Una vez una persona me dijo que no le gustaba el libro de texto pautado para el curso, porque *había que leer para poder entender un capítulo* (en serio).
- Entienda lo que está haciendo: Importante es enterarse de que está pasando y que estas haciendo. Escribir un código que bajaste de un sitio y no saber que hace específicamente sólo crea lagunas. Debes de analizar y entender a profundidad que está pasando, por qué pasa y por que cuando borro una línea el código funciona.
- Domine el paradigma de orientación a objetos: He tenido alumnos que llegan a programación II y todavía tienen deficiencias críticas en este sentido. Para poder ser exitoso en cualquier lenguaje orientado a objetos es indispensable

saber distinguir entre un objeto , una referencia y una clase. Aunque parezca muy sencillo, se sorprenderían al saber la cantidad de personas que no saben esto , están en el limbo y cuando tienen que acceder a un objeto en específico están totalmente perdidos.

Es importante tener la capacidad de poder visualizar el sistema como un conjunto de objetos trabajando juntos entre sí. En muchos casos he visto códigos que NO visualizan el sistema como tal y lo que parece es un más un código estructurado, y hacen clases como con nombres como "*Funciones, Calcular, AgregarUsuario*". Les aconsejo que antes de hacer una clase se pregunten a si mismos como se escucharía un objeto de esa clase, ¿tendría sentido?. Ejemplo: Tiene sentido "Yo soy una calculadora", o "Yo soy un Recibo" , o "Yo soy un reporte", pero hay cosas que no tienen sentido como "Yo soy un Funciones" , o "Yo soy un Calcular", o "Yo soy un Usuarios (en plural)".

- Preste atención a los errores: Cuando su programa le de un error de compilación o en tiempo de ejecución LEA el error, y si son varios léalos desde el primero hasta el último. Se sorprenderían de saber la cantidad de principiantes que NO leen los errores y desde que ven un stacktrace , solo dicen AH NO FUNCIONA, y cuando les pregunto que cual error le está dando me dicen: "na' que no funciona". La máquina virtual de java es muy explícita dando mensajes de error, En caso de leer el mensaje de error y no poder deducir el problema busque el mensaje el google.
- No se acostumbre al copy & paste!: He tenido alumnos que han sido exitosos en hacer tareas. No obstante para "*ahorrar tiempo*" en vez de empezar desde cero hacen un copy y paste del "esqueleto" de un código anterior. En el examen parcial que no pueden usar código anterior , se les "abre el pecho" porque no son capaces de tirar una línea sin hacer copy y paste. Está demostrado que la repetición es un recurso didáctico efectivo. Si usted empieza cada código desde cero verá que ese conocimiento permanece.
- Practique! : Una frase célebre de Bjarne Stroustrup (Creador de C++) es: "*La única forma de aprender a programar es programando*" y esto sin lugar a dudas es una realidad y cualquier personaje destacado en el mundo de la programación lo confirma , como el caso de Richard Stallman. Practique con el código que hacemos en clase, córralo, modifíquelo, corralo con las modificaciones a ver que pasa. Intente hacerlo desde cero sin mirar, intente hacerlo mejor. Estas son las actividades que hacen que los conocimientos de programación se "*queden pegados en tu cerebro*".

- Adivinar nunca es un método efectivo: He conocido personas que cuando un código no funciona en vez de leer, reintentar y luego buscar en google lo que hacen es intentar adivinar sin ninguna base lógica. Recuerdo una vez que intentaba correr en clases una clase que estaba en un paquete pero tenía problemas de classpath. Escribí en la consola `java edu.itla.agenda.Sistema` y obviamente una excepción de tipo `NoClassDefFoundError` se arrojó y alguien me dijo "Intente sustituir los puntos por espacios a ver si funciona" , obviamente una sugerencia carente de base , meramente basada en la adivinanza y no basado en el análisis crítico que debe de desarrollar un programador exitoso.
- Persevere!: En ocasiones he visto como alumnos se sienten desmotivados pues su código no les funciona. La programación se trata de retos. Si todo compilara desde el primer intento estuviera yo ahora mismo estudiando otra profesión, pues si programar fuera algo rutinario que no representaría ningún nuevo reto y no despertaría pasión. Me he topado con casos en que he tenido que hacer más de 80 intentos (sin exagerar) resolver un problema, y con entusiasmo, esmero, pasión y perseverancia se logra.

Espero que estos consejos les sean de utilidad, practique y diviértase mucho programando!