**EJERCICIO 1.**

* No tiene límite de usos, es decir, se pueden usar en cuantas estaciones de trabajo se quiera.
* Es gratuito, no tienes que pagar por la licencia.
* Se tiene acceso a su código, con lo cual se puede modificar y distribuir a tu gusto.
* Ejemplos: Linux, Libre Office, VLC.

**EJERCICIO 2.**

Yo creo que se debe a que no se cuestiona bien la función del programa/proyecto que se quiere desarrollar. No hay una buena comunicación y entendimiento con el cliente, lo que llega a que luego cuando se ponga en marcha no sea efectiva o se deba de modificar debido a que el programa no hace lo que el cliente quería.

**EJERCICIO 3.**

Yo creo que sería la etapa de Análisis por lo dicho en el ejercicio anterior. Un buen análisis del programa conlleva a que se deba de modificar lo menos posible y a ponerlo en marcha cuanto antes para que el cliente esté lo más satisfecho y no lo cancele por continuos retrasos.

**EJERCICIO 4.**

Es falso. Durante la creación de un programa informático pueden surgir desde dudas o modificaciones por parte del cliente, llegando incluso a que no se exprese concretamente y no se haga lo que pide desde un principio.

**EJERCICIO 5.**

Requisitos funcionales:

* Que escanee el código de barras de un producto.
* Que haga la cuenta total del costo de la compra de un cliente.
* Que se pueda meter la cantidad de productos que se han recibido.
* Que haga las nóminas de un trabajador.
* Que haga la caja total al cierre del establecimiento.

Requisitos no funcionales:

* Que el programa pueda usarse en Windows o Linux (o en el sistema operativo que tenga el establecimiento)
* Que posea una base de datos con su correspondiente sistema gestor para almacenar los datos de los trabajadores y productos que tiene.

**EJERCICIO 6.**

Mantenimiento adaptativo: Se da cuando se producen cambios en el sitio donde se va a implementar el programa. Ya sea porque han cambiado de software o porque ha cambiado la empresa. Por ejemplo, se cambian las estaciones de trabajo de Windows a Linux o por ejemplo un trabajador obtiene otros privilegios que antes no tenían cuando ejecutan el programa.

Mantenimiento correctivo: Se da cuando aparecen errores que no se habían detectado. Por ejemplo, un trabajado descubre un bug cuando le da a un botón.

Mantenimiento perfectivo: Se da cuando se quieren realizar cambios para mejorar el programa. Por ejemplo, se quiere dar más seguridad a los mensajes que se envían a través de programa y se cambia el tipo de cifrado de estos.

Mantenimiento preventivo: Se da cuando se quieren añadir o modificar comentarios en su código. Por ejemplo, se ha añadido una nueva función en el programa, con lo cual, añadimos un comentario para que el que vuelva a ver el código sepa cómo va.

**EJERCICIO 7.**

Sí, debido a que si por algún motivo el proyecto se deja a medias o se destina a otro programador, pueda entender fácilmente el código para continuar con él.

**EJERCICIO 8.**

1. Análisis lexicográfico: Se buscan palabras reservadas, operaciones, caracteres de puntuación, etc.
2. Análisis sintáctico-semántico: Se revisa la coherencia.
3. Generación de código intermedio.
4. Optimización del código.
5. Generación de código.
6. Enlazador del código con las librerías.

