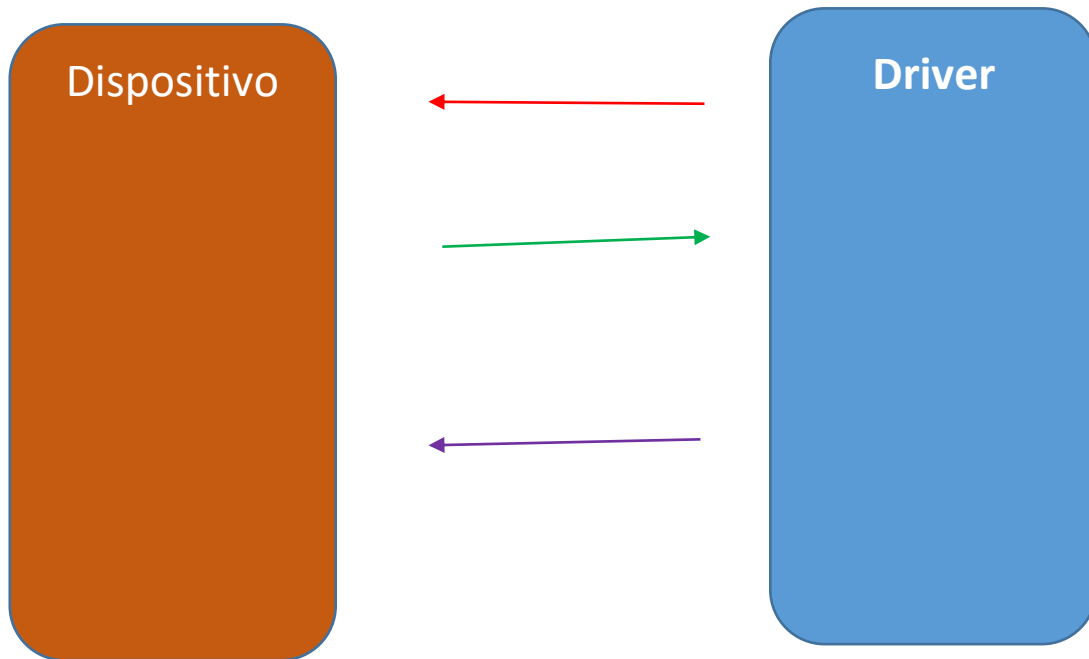


Importancia y creación de un protocolo

El protocolo es una parte especial en la creación de cualquier programa o dispositivo, es el encargado de decirle al dispositivo o programa lo que tiene que hacer en base a lo que el usuario envíe e indicarle al usuario si se realizó la acción o no y en el caso que no se realice por qué no se pudo hacer.. El protocolo en si no es un programa aparte y no hay una forma fija de crearlo ya que siempre será distinto dependiendo del uso que queramos darle



En la imagen anterior el protocolo se ve representado con la flechas. El driver recibe la información de usuario y mediante el protocolo se le envía al dispositivo la tarea que debe realizar.

Se puede pensar en el protocolo como un switch ya que cumplen la misma función y en muchos casos el protocolo se puede implementar como uno.



Comunicación PCI-Protocolo-Driver

La base principal de la comunicación entre el dispositivo (en este caso el PSI) y el driver es el protocolo, como se dijo antes este es la comunicación entre ambos, pero su creación tiene varias reglas que se deben seguir si se quiere crear de la mejor manera.

- **Prevención de datos no deseados:** Como anteriormente se mencionó el protocolo es muy similar a un switch y como tal se tiene que pensar en una forma de manejar datos que no se desean.

Ejemplo:

```
switch(indicador) {  
    case x:  
        // código de caso x  
        break;  
    case y:  
        // código de caso y  
        break;  
    default:  
        // Maneja el resto de datos  
}
```

Como se muestra en el ejemplo anterior el switch posee un caso default e cual maneja todos los datos que no sean de utilidad para el programa.

- **Manejo en orden de los datos y procesos:** A la hora de recibir los datos no solo importa si el dato es válido, además importa el orden en el cual son recibidos, pensándolo como una transacción en un cajero se puede ejemplificar mejor: Si se desea sacar 1000000 colones de un cajero se debe seguir una serie de pasos siendo el primero ingresar la tarjeta, el usuario puede intentar darle otras indicaciones al cajero tocando los botones sin embargo este no permitirá ninguna acción hasta que se inserte la tarjeta de la manera debida.



Ejemplo Final:

Tomando el ejemplo del cajero automático un protocolo básico se puede formar de la siguiente forma.

Operaciones a realizar en el orden adecuado:

1. Insertar la tarjeta
2. Poner el Pin
3. Definir si es un retiro o una consulta
4. El tipo de cuenta
5. El monto a retirar
6. Si desea un recibo
7. Si desea realizar otra transacción

Ahora cada operación tiene que seguir las 2 reglas anteriores, debe de ser imposible definir el monto a retirar sin haber insertado el pin o poner el pin sin haber insertado la tarjeta. Además cada operación tiene sus propias reglas, por ejemplo si marca consulta muchos pasos pueden ser evitados o ser totalmente distintos, además debe de ser imposible para el usuario ingresar un dato incorrecto, por ejemplo una letra “a”.

Viendo la lista de operaciones se puede notar que la operación que define como se va a comportar el dispositivo es la 3, dependiendo de la selección el cajero se comporta de forma distinta, esto se puede representar como un switch básico.

```
switch(Selección) {  
    case retiro:  
        // Continúa con el proceso del retiro  
        break;  
    case Consulta:  
        // Continúa con la consulta.  
        break;  
    default:  
        // Maneja el resto de datos no deseados  
}
```



Suponiendo que se elige la opción de retiro viene otra consulta que corresponde al tipo de cuenta la cual se puede representar como otro switch anidado.

```
case retiro:
    switch(Selección) {

        case Ahorro:
            // Continúa con la cuenta de ahorros.
            break;

        case Normal:
            // Continúa con la cuenta normal.
            break;

        default:
    }
```

Por último se pide el monto y si desea recibo, en el caso del monto se debe revisar que el monto no exceda la cantidad máxima ni sea mejor que la cantidad mínima además de verificar si el cajero tiene el dinero necesario para la transacción, esto se puede representar como una serie de if, que en el caso que suceda alguna excepción se lo indique al usuario con un mensaje.

El caso del recibo se puede representar como un bool que si se encuentra en True al final de la transacción esto signifique que se debe generar un recibo para el usuario.

Como se puede notar el protocolo es la base de la comunicación usuario (driver) y el dispositivo, es el encargado de mandar las instrucciones entre ambos y saber cómo manejarlas, manejar todos los problemas que se puedan presentar y evitar que estos sucedan, un buen protocolo es esencial para la creación de un dispositivo fácil de usar y con la mejor cantidad de errores posibles.

