**VI. Prácticas**

**Requisitos:**

**Req-SW1:** El sistema debe detectar un corto a batería y a tierra.

**Req-SW2:** Se debe tener una función que sea llamada para reportar si se detecta un corto a batería o a tierra usando una enumeración dónde: 0 = Sin corto; 1 = Corto a Bateria y 2 = Corto a tierra.

**Req-SW3:** La función deberá llamar una interface mandándole como parámetro un bool en true si la falla está presente y un false si la falla no está presente.

**Implementación Actual:**

typedef enum Corto\_Tag

{

SIN\_CORTOS = 1,

CORTO\_BATERIA = 6,

CORTO\_TIERRA = 7

}Corto\_T;

void Interface\_falla(bool hay\_falla)

{

/\*Imprime el estado de la falla\*/

printf("¿La falla esta presente?");

printf(hay\_falla? "true" : "false");

}

void Reporte\_de\_falla(Corto\_T dato)

{

bool falla = false;

if (SIN\_CORTOS == dato)

{

falla = true;

}

Interface\_falla(falla);

}

**Reporte del defecto:**

La función **Reporte\_de\_falla** no imprime el estatus de la falla cuando se llama la función con un 1: **Reporte\_de\_falla(1);**

**Corrección de funcionalidad:**

Siguiendo el diagrama de flujo de la Figura 5 Flujo a bloques para un cambio por defecto, describe y ejecuta los pasos.

1. ¿Es un “Bug”? ¿Por qué?

R: Si. Porque existe malfuncionalidad debido a la enumeración incorrecta y la comparación en la función **Reporte\_de\_falla**

1. Dale un nombre a tu tarea y regístrala en una hoja.

R:

|  |  |
| --- | --- |
| Nombre de la tarea | Task\_Reporte\_falla\_issue |
| Solucionador del problema | Rafael Iglesias |
| En que versión se detecto | Versión 0 |
| Estatus | Submitted |

1. Si tuvieras una herramienta para el control de cambios de software, ¿cuáles serían los pasos que deberías seguir?

R:

1.- Ir a la última versión funcional

2.- Crear un Branch y apartir de ahí trabajar en la solución

3.- Integrar cambios.

1. Has los cambios necesarios para corregir la funcionalidad.

R: typedef enum Corto\_Tag

{

SIN\_CORTOS = 0,

CORTO\_BATERIA = 1,

CORTO\_TIERRA = 2

}Corto\_T;

void Reporte\_de\_falla(Corto\_T dato)

{

bool falla = false;

if (SIN\_CORTOS ¡= dato)

{

falla = true;

}

Interface\_falla(falla);

}

1. Realiza una inspección al código para validar su funcionalidad.

N/A

1. Documenta los cambios en la hoja donde creaste tu tarea considerando todos los ítems mencionados en el Cap. V paso 5.

R:

|  |  |
| --- | --- |
| Nombre de la tarea | Task\_Reporte\_falla\_issue |
| Solucionador del problema | Rafael Iglesias |
| En que versión se detecto | Versión 0 |
| Cambios | Se realizaron cambios en la enumeración enum Corto\_Tag y en la función Reporte\_de\_falla(Corto\_T dato) |
| Estatus | Implementado |

1. Cierra la tarea con una leyenda de “Implementado”.

R: Respuesta en la anterior

1. Escribe tus conclusiones.

R: Este tipo de herramientas son muy útiles para el control de cambios en sw.