

CLASE 5 (tomar lista)

Partición de Equivalencias y Herramientas

Hablamos un rato de los Casos de Prueba de “El Provinciano”

- ¿Qué casos se les ocurrieron?
- ¿Pudieron probarlos?
- ¿Cómo reportaron los bugs / errores?

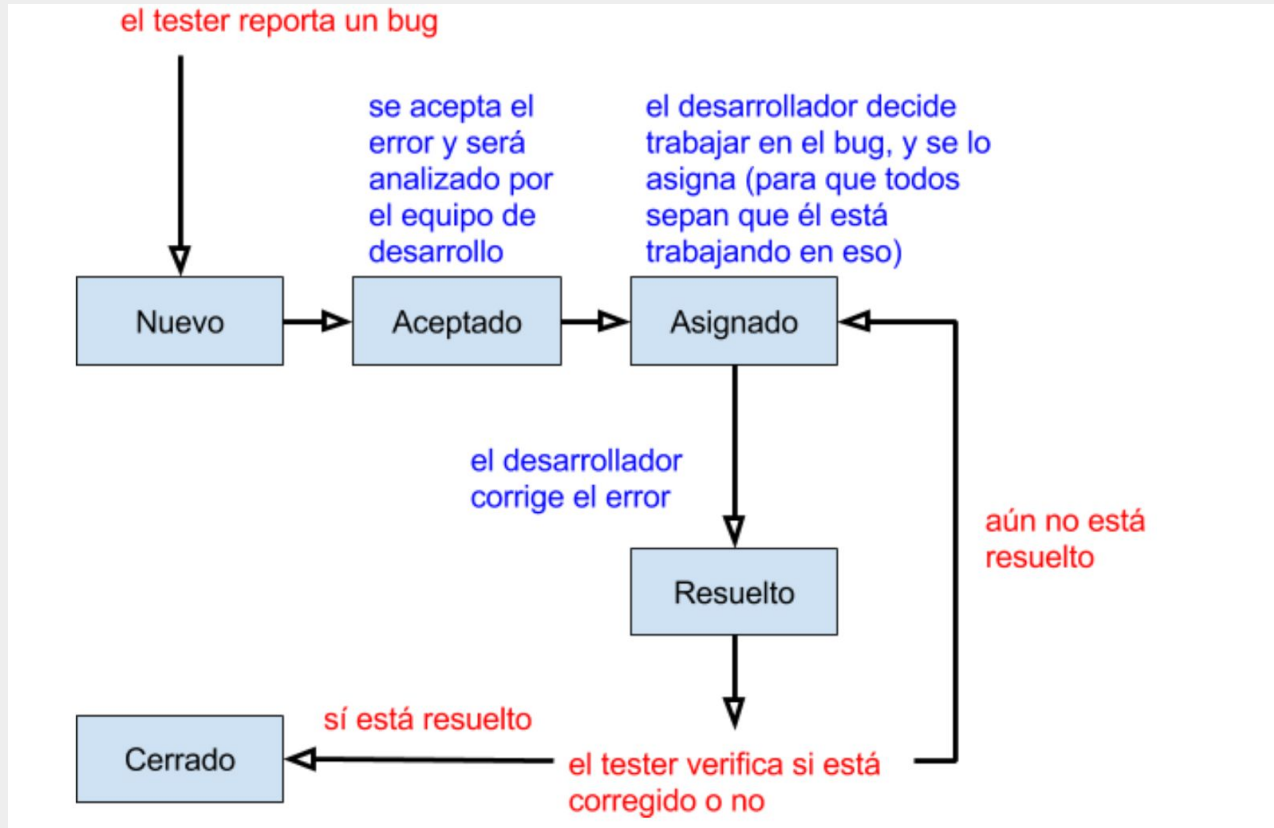
Ciclo de Vida de un Bug

El flujo más típico de los incidentes (o bugs) es el que se representa en el siguiente esquema. Considera que en rojo aparecen las acciones que hace el tester y en azul aparecen las acciones que hace el desarrollador. Luego verás en las cajas las distintas situaciones en las que se encuentra un bug, las cuales son:

Ciclo de Vida de un Bug

- Nuevo: un bug recién creado por un tester.
- Asignado: un bug que un desarrollador decidió corregir.
- Resuelto: un bug que un desarrollador corrigió y está esperando que un tester verifique que quedó correctamente corregido.
- Cerrado: un bug que ya fue resuelto y verificado por un tester, con lo cual ya no hay más para hacer.

Ciclo de Vida de un Bug : Gráfico explicativo



Clases de Equivalencias

Dado que probar todo es imposible, y en realidad del universo de entradas posibles vamos a poder probar solo un subconjunto, queremos elegir aquel subconjunto que tenga la mayor probabilidad posible de encontrar errores.

El desafío: ¡probar lo más posible con el menor esfuerzo posible! :)

Técnica de diseño

Partición en Clases de Equivalencia es una técnica de diseño de casos de prueba de Caja Negra.

Pasos de la técnica:

Paso 1: Identificar las clases de equivalencia

Para identificar las clases de equivalencia, se consideran los valores de entrada y los valores de salida del programa bajo prueba.

Se definen 2 tipos de clases de equivalencia: *clases válidas* y *clases inválidas*. Las cuales corresponden a entradas válidas e inválidas, respectivamente.

Para empezar:

- Identificamos las variables existentes y sus posibles valores.
- Identificamos las clases de equivalencia.

¿Cómo identificar las clases de equivalencia?

Si la condición de entrada corresponde a un rango de valores (por ejemplo: “un número entre 1 y 999”) → se define 1 clase de equivalencia válida (CEV) y 2 inválidas (CEI). En el ejemplo:

- Clase **Equivalencia Válida**: Números Iguales o Mayores a 1 // Números iguales o menores a 999
- Clase **Equivalencia Inválida**: valores Menores a 1 // Valores Mayores a 999

Si la condición de entrada corresponde a un valor específico (por ejemplo: “el primer carácter tiene que ser una letra”) →

- Clase **Equivalencia Válida** (una letra)
- Clase **Equivalencia Inválida** (no es una letra).

Técnica de diseño

Paso 2: Definir los casos de prueba

Procedimiento:

- Definir casos de prueba hasta contemplar todas las clases de equivalencia válidas (una caso de prueba para cada clase de equivalencia válida) Intentando incluir la mayor cantidad de clases válidas en cada caso de prueba (de modo de minimizar la cantidad de casos de prueba).
- Para las clases de equivalencia inválidas, definir un caso de prueba por cada clase de equivalencia inválida

Issue Tracker

Explicamos brevemente ***Jira***:

Jira: <https://test-unahur.atlassian.net/jira/software/projects/TU/boards/1>