Entonces ya vimos pruebas de caja negra, vamos ahora a repasar concepto de pruebas de caja blanca, decíamos que las pruebas de caja blanca son pruebas estructurales porque revisa la estructura del código, también son conocidas como clase dos porque digamos como si fuera una caja transparente no, en caja blanca después se puede ver internamente qué lo que hace, se prueba lo que el software hace, se evalúa el resultado aprovechando el conocimiento interno del código para dirigir la prueba, aprovechamos lo que sabemos del código para poder dirigir la prueba hasta donde nosotros queremos, estas pruebas de caja blanca no garantizan el cumplimiento de las especificaciones funcionales, son pruebas complementarias a las funcionales, las funcionales son las de caja negra, estas son complementarias, en cada caso de prueba debe probar esto un ejemplo de todos los posibles, por ejemplo si la función contiene una condición "if/else" si pasa esto entonces, si no sabe otra cosa, se deben hacer dos casos de prueba, cuando hay una bifurcación en el código se deben hacer dos casos de pruebas, si aparece un "or" un "and" es decir esto y además esto otro, si se cumple esta condición o esta otra condición se deben generar dos casos pruebas para la misma condición y así, el tema es poder cubrir todo toda la lógica por donde va el código, que se acuerdan que habíamos hecho un flujo no, entonces el código puede ir para un lado y para el otro bueno hay que tratar de cubrir todo el árbol, este tiene una estructura de árbol, con su nodo principal y su rama distintos nodos, distintas ramas que sobre de ahí, bueno la idea es poder cubrir todo, todo, todo, todo ese lugar el desarrollador al conocer la estructura interna puede dirigirse directamente a las partes del código más riesgosas generalmente realizado este este test por el desarrollador o con el desarrollador, entonces acá tenemos el ejemplo el flujo de lógica, ahí tenemos un programita que dice bueno este hace un input de A y B estamos leyendo la parte izquierda de la pantalla e input A y B, ingresa dos valores al sistema cada valor es desde 1 hasta 100, ingresamos los números válidos de 1 hasta 100, pregunta si el primer número el A es mayor a 50 entonces hace la sentencia 1 y si B es menor a 25 entonces hace la sentencia 2 y si no hace la sentencia 3 y si no se cumple en de arriba el mayor a 50 hace la sentencia 4, esto es lo que está haciendo este este código, entonces si vamos al gráfico, a la derecha, tenemos la representación de esta misma, preguntas el nombre siempre está determinando que hay una pregunta entonces el nombre es una pregunta, si A es mayor a 50, sí, si esto va a pasar cuando, va a pasar el 90% de las veces y entonces vamos vamos para este lado si sí, si no vamos para el otro lado si aquí no cual, esto va a pasar el 10% de las veces, entonces se dividen o se dilata en el flujo, una más grueso porque van muchas más opciones por ahí, otro más finito porque van menos opciones por ahí, aparece cuatro cuentas van a ejecutar, nada más que mil veces, si si nosotros ponemos todos los números no, estamos haciendo una estadística para si nosotros ingresamos por ejemplo tenemos un número A que puede ir de 1 a 100, el número B que puede ir de 1 a 100 estos son 10 mil combinaciones, entonces esos 10 mil combinaciones hay 1000 que van a ir por un lado y 9000 por otro y después pregunta si es menor a 25 entonces también hay este esa otra pregunta, un 75% que va por un lado y un 25% que no para el otro, se multiplica por algunos para saber la cantidad de casos se multiplican por lo que sintieron de la condición anterior, por que era un 90%, entonces tenemos 2.250 casos por un lado y 6.750 casos por el otro, acá tenemos como iría el flujo no adentro del sistema, si nosotros ingresamos números algunos, cubrimos todo el espectro, sabemos que hay 6.750 veces que va a ir por un lugar 2.250 por otro y 1.000 veces que va a ir por otro esto es importante saber para saber dónde con más probabilidad va a ir el sistema, este este es el nivel más bajo del testeo por cobertura estadística que cada sentencia del software testeado debería ser ejecutada por lo menos una vez, ya me quedaría tranquilo si sé que cada cosa que está escrita en el código por lo menos pasa por ahí una vez el sistema y una vez se ejecute, entonces qué pasa, decíamos que aunque se realice esta cobertura total cubrimos todo el código, se pueden perder muchos caminos posibles, como es esto, a veces no es sencillo realizar la cobertura total, por