Bien entonces la prueba de caja negra se acuerdan que son estas pruebas que nosotros decíamos que ¿porque las llamamos caja negra? Porque no sabemos qué pasa adentro de la caja, no sabemos qué es lo que está haciendo el sistema, sabemos que nosotros le damos una entrada y tiene una salida pero digamos que hace internamente no sabemos, si, sí lo que sabemos es cómo digamos que funcionamiento tiene que tener ya que si le damos esta entrada va a tener que tener esta salida eso sí lo sabemos, entonces son pruebas funcionales o inducidas por datos, que tienen que ver de acuerdo a la entrada es la salida, prueba lo que el software debería hacer, toma como punto de partida la definición del módulo a probar, hay que leer la documentación en función de esta documentación entender si le damos cierta entrada qué salida va a tener y probar que digamos esto sea cierto, sea verdadero, se limitan a que el tester pruebe con datos de entrada y estudie como salen, sin preocuparse por lo que ocurre en el interior, o lo anterior se suelen llamar también pruebas de entrada-salida es sinónimo, las pruebas exhaustivas de caja negra son imposibles de realizar en la mayoría de los sistemas, cuál es la solución, acotar las pruebas, cómo se acotan las pruebas, seleccionando subconjuntos de las pruebas que permitan cubrir un conjunto extenso de otros casos de pruebas posibles, si yo con una unas pocas pruebas puedo cubrir una gran variedad de salidas, en ese caso digamos ahorro esfuerzos y es difícil encontrar un conjunto de pruebas adecuados, con el tamaño adecuado para abarcar el dominio y maximizar la probabilidad de encontrar errores, o sea, si yo estoy probando tengo que ingresar colores y por cada color puede pasar algo distinto bueno sabemos qué colores hay muchísimos pero bueno vamos a probar con los colores más comunes, más importantes, pero tratando de abarcar pongamos uno en la gama del amarillo-naranja no pongamos todos en la gama del rojo y nada en la gama del azul y tratemos de tener todo, unir todo el espectro, entonces este y para poder ejecutar el proceso de testing con cada elemento del conjunto minimizando el costo del mismo, entonces cómo es esto, las pruebas de caja negra si, tienen unas técnicas de derivación de casos de prueba, es decir, cómo podemos nosotros armar los conjuntos de datos para poder hacer las pruebas, entonces acá tenemos 4 vamos a ver 4 técnicas, la clase equivalencia, las condiciones de borde, el ingreso de variables de otro tipo y la conjetura de errores. La clase de equivalencia es una agrupación de datos de entrada y resultado de salida, todos los miembros de cada grupo están relacionados, cada clase es una partición de equivalencia en la que el sistema se comporta de la misma forma para cada miembro de la clase, la prueba de un valor representativo de cada clase es equivalente a la prueba de cualquier otro valor, estas pruebas son llamadas pruebas por partición de equivalencia o pruebas basadas en sus dominios, entonces ahora vamos a dar un ejemplo que se va entender, pasos a realizar, hay que identificar las clases equivalencias y hay que definir los casos de prueba, entonces aca vamos, se va a terminar entender la idea, se divide cada condición de entrada en dos grupos, las clases válidas y las clases inválidas, entonces si yo sé que un número de sucursal de un banco, por ejemplo, si, sabemos que está entre 1 y 99 la clase válida es si el número es mayor a 1 y menor a 99, si la clase es menor a 01 es inválida, si la clase es mayor a 99 es inválida, sabemos que tiene que estar entre 1 y 99, no, se entiende, lo que está fuera de esos márgenes tenemos acotado de lo que está por fuera, bueno, lo que está por fuera es invalido, entonces nosotros separamos de esa manera, por ejemplo, si tenemos las siguientes opciones de entrada de datos una condición de entrada especifica un rango de valores, en sucursal entre 1 y 99, si una condición de entrada especifica un conjunto de valores y existen razones para que creer que el programa no se trata de distinto, este es otro caso, entonces cuál sería la clase válida y cuál sería la clase inválida, por ejemplo, si yo tengo que agregar un tipo de documento entonces la una clase valida es dni, es una clase válida, cedula de identidad es una clase valida, pasaporte es una clase valida, todo lo que no son válidos son clases invalidas, cualquier otra cosa que escriba es una clase inválida, entonces esto me sirve a mí para separar cuál es una entrada

válida de cuál es una entrada inválida, entonces en el caso anterior, por ejemplo, si yo tengo que hacer una prueba acá, tengo tres clases una clase válida y dos clases inválida, voy a probar con un valor que tomo de la clase válida, no se pongo el 55, que está entre 1 y 99, tomo un valor de una clase inválida, por ejemplo el 0 y toma un valor, otro valor de la segunda clase invalida que por ejemplo de 180, entonces pruebo con esos valores tengo 3, ya tengo tres pruebas para hacer, pero tengo tres pruebas, el dominio de esto es muy grande porque son los números, son todos números, o sea, yo podría poner cualquier número digámos me puedo cansar haciendo pruebas con millones de números, lo que pasa que lo que indica la lógica es que nosotros con hacer tres pruebas es suficiente, no necesitamos probar más y acá lo mismo, acá vamos a probar algo que sea válido y algo que se invalidó, entonces para hacer dos pruebas estamos bien, siempre se hacen la misma cantidad pruebas que clases, entonces acá hay otra otra categoría que es la categoría debe ser no así una condición de entrada específica que debe ser por ejemplo el primer carácter debe ser un dígito, entonces la clase válida sería si el primer carácter es un dígito y la clase inválida y el primer carácter es distinto de un dígito, una letra por ejemplo, entonces ahí tenemos dos clases vamos a probar con un valor de una clase con un valor de otra clase, siempre tomamos un valor que represente la clase, un valor que represente la clase probamos con ese valor y tomamos otro valor que represente la clase inválida y probamos con eso, si queremos que los elementos de una clase de equivalencia no son tratados en forma idéntica debemos dividir la clase en clases menores, por ejemplo las sucursales de Capital Federal son entre las 100 y las 130, o sea, que nosotros estamos diciendo que las sucursales entre la 1 y la 99 pero la de Capital Federal son distintas, Capital Federal van entre la 100 ya 130, bueno listo lo separamos en otra clase, bien ahora vamos a hablar de condición de borde, hasta ahora estuvimos hablando de clase de equivalencia, ahora vamos a hablar de condición de borde, para qué sirve todo esto que les estoy contando, sirve porque ustedes van a tener que diseñar casos de prueba y cuando diseñen casos de prueba van a tener que decir de qué manera van a probar, si ustedes tienen que dar el ingreso en un formulario donde hay un montón de datos, si, datos para tramitar el dni y ustedes tienen que cargar datos de las personas, por ejemplo, o lo que sea no, cualquier formulario que tengan que cargar donde se pida el ingreso de datos, se pide información, ustedes tienen que analizar los datos de entrada posibles para poder probar digamos las menos veces posibles y con la menor cantidad de pruebas posibles para poder cubrir un amplio espectro de opciones dentro del sistema, entonces por eso sirve esto, para que ustedes puedan diseñar convenientemente sus casos de prueba. Vamos a hablar de la condición de borde, es una variación de la técnica de partición de equivalencia, focalizada en los bordes de cada equivalencia por arriba y por abajo, los casos de pruebas que exploran las condiciones de borde producen mejor resultado que aquellas que no lo hacen, por ejemplo con esto que decíamos de la sucursales de entre 1 y 99 la clase valida, o sea, hay una clase valida que si pongo sucursal 1 es válido si pongo sucursal 99 es válido porque estaba entre 1 y 99 incluyendolo, entonces nos vamos a los bordes, nos vamos a los bordes de la clase, cual es si la clase válida es entre 1 y 99, bueno vamos a 1 y 99 voy a probar con un 1 y 99 que son los bordes y despues voy a probar las clases inválidas, decíamos también era menor que 1 y mayor a 99 esto sería probar también en los bordes sucursal 0 sería uno de los bordes sucursal 100 es otro de los bordes, se prueba siempre en los bordes, esto puede servir para detectar errores de programación porque cuando el programador a este a veces pone un en vez de poner mayor igual o menor igual pone un mayor, se olvida al igual, entonces ahí suceden problemas entonces por eso dice que los casos de pruebas que exploran condiciones de borde producen mejor resultado que aquellos que no lo hacen, bien si la condición de entrada puede ser un rango de valores, seleccionar los casos válidos para los extremos del rango y los inválidos para para los valores siguientes a los extremos, ven esto es lo que acabamos de ver, aplicar el mismo criterio para los datos de salida, lo mismo pasa con la otra salida, la cantidad de niñas que podría tener un reporte de salida, trabaja de la misma forma, prestar especial atención a los archivos/tablas,

vacío primer registro/fila, último registro digamos si hacemos una salida en una tabla o una impresión bueno también tenemos que tener en cuenta que la salida tenga una cantidad de registros este digamos, que dentro de los bordes o por fuera de los bordes, bien hay otro tipo de prueba caja negra que son los valores de otro tipo, cuanto yo hago pruebas ingreso en un campo valores de otro tipo, por ejemplo si se está esperando un valor numérico ingreso en algún alfabético, si está esperando un valor alfabético, si el sistema está esperando que les carguemos un valor alfabético podemos ingresar un valor numérico, podemos ingresar combinaciones de ambos o podemos ingresar, por ejemplo fechas erróneas 31 de febrero de 1990 dice acá es una fecha errónea, bien este es un tipo de prueba ya esto lo venimos viendo de las primeras clases, o sea, como nosotros vamos a hacer para probar para probar los campos de ingreso, la combinación de los datos de entrada es la que produce una clase valida o inválida, ejemplo podría ser que si el estado civil es divorciado los datos del conyuge se pueden ignorar y por otro lado, dentro de las pruebas de caja negra tenemos otra técnica que se llama conjetura de errores, que se trata que nosotros sospechamos que algo puede andar mal, enumaremos una lista de errores posibles o de situaciones propensas a tener errores y creamos casos de prueba basados en estas situaciones, como sería esto, porque acá la creatividad juega un papel clave, no hay una técnica para la conjetura de errores, es un proceso intuitivo que se basa en la experiencia, entonces cómo es esto, yo sospecho que hay que en este módulo puede haber errores, que puede haber errores con ciertos datos, cuando cuando sucede esto digamos, que es lo que me da la pista de que esto puede ser, bueno si parte de ese módulo fue hecho a las apuradas, rapidito rapidito que tenemos que terminar, el programador programa rápido, practicamente no se testea y sale el desarrollo, bueno cuando llega a nosotros ya vamos a tener esa información de que fue hecho las apuradas, sabemos eso, y bueno ahí sospechamos que algo puede andar mal, si fue modificado por varias personas en distintos momentos, si hay muchas personas involucradas y una tomó una cosa y otros toma otra esto también puede llevar a errores, si tiene estructuras anidadas, condiciones compuestas, o sea, si es la lógica muy intrincada, muy difícil este también bueno, eso es algo que es propenso a errores y si fue armado por la tecnica de “copypaste” de otros componentes, o sea, a ver yo tengo que hacer el alta de clientes y ya tengo el alta de proveedores bueno lo copió, copio el alta de proveedores en al de clinte y les cambió algunos rotulos, si hago eso a veces hay cosas que quedan mal, bien hasta acá entonces lo que les quería comentar acerca de las pruebas de caja negra y ahora vamos a hablar un poco sobre las pruebas de caja blanca.