Aclaración: para aquellos métodos que se les haya realizado Caja Negra y que posean más de un parámetro de entrada, se considerara para su evaluación de a un parámetro por vez. Esto es, si por ejemplo se tuviera lo siguiente:
public void cajaNegra(“TIP000000”,2,”hola”)
Primero se determinara la clase correcta que involucra todos los parámetros, y luego, tanto para los valores límites como para las clases erróneas, se tomará de a un parámetro. Siendo así que se modificará sólo el parámetro que está siendo evaluado en ese momento y todos los demás quedarán con el mismo valor que tenían cuando se indicó la clase correcta.
Esto se realiza para evitar así las incontables combinaciones posibles que resultarían si no se realizaran los análisis individualmente.
El mismo criterio se tomará al realizar el testeo unitario de caja negra.

Constructor

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Lista de materiales	{lista != null} 1	
Descripción	{descripción != null} 2	

Batería de pruebas

	Entradas	Salidas	Clases cubiertas	Valores Límite	Salidas
Clases correctas	{lista != null, descripción != null}	nueva instancia de la clase	1,2		

Constructor

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Pedido	{pedido != null} 1	

Batería de pruebas

	Entradas	Salidas	Clases cubiertas	Valores Límite	Salidas
Clases correctas	{pedido != null}	nueva instancia de la clase	1		

agregarObservacion()

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Observacion	{obs != null} 1	
Tema	{obs.tema != null} 2	
Legajo	{obs.legajoEmpleado = LEGXXXXXX && 0<=X<=9} 3	
Texto	{obs.texto != null} 4	
FechaObservacion	{obs.fechaObservacion != null} 5	

Batería de pruebas

	Entradas	Salidas	Clases cubiertas	Valores Límite	Salidas
Clases correctas	{obs != null,obs.tema = "prueba", obs.fechaObservacion != null, obs.legajoEmpleado = LEGXXXXXX, obs.texto != "es una observacion"}	observación agregada	1,2,3,4,5		

aceptarPedido()

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Aceptar	{pedido.fechaPedidoAceptado != null, pedido.fechaPropuestaProduccion != null, pedido.fechaDefinitiva != null} 1	{pedido.fechaPedidoAceptado = null} 2.1 {pedido.fechaPropuestaProduccion = null} 2.2 {pedido.fechaDefinitiva = null}2.3

Batería de pruebas

	Entradas	Salidas	Clases cubiertas	Valores Límite	Salidas
Clases correctas	{pedido.fechaPedidoAceptado != null, pedido.fechaPropuestaProduccion != null, pedido.fechaDefinitiva != null}	pedido aceptado	1		
Clases erróneas	{pedido.fechaPedidoAceptado = null}	El pedido no esta listo para ser aceptado	2.1		
	{pedido.fechaPropuestaProduccion = null}		2.2		
	{pedido.fechaDefinitiva = null}		2.3		

Constructor

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Mensaje	{mensaje != null} 1	
Lista de materiales	{faltantes != null}2	

Batería de pruebas

	Entradas	Salidas	Clases cubiertas	Valores Límite	Salidas
Clases correctas	{"es una prueba", faltantes != null}	nueva instancia de la clase	1,2		

Constructor

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Mensaje	{mensaje != null} 1	

Batería de pruebas

	Entradas	Salidas	Clases cubiertas	Valores Límite	Salidas
Clases correctas	{"es una prueba"}	nueva instancia de la clase	1		

agregarNuevo()

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Lote	{nuevo != null} 1	

Batería de pruebas

	Entradas	Salidas	Clases cubiertas	Valores Límite	Salidas
Clases correctas	{nuevo != null}	agrega Lote	1		

borrarLote()

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Lote	{lot != null} 1	

Batería de pruebas

	Entradas	Salidas	Clases cubiertas	Valores Límite	Salidas
Clases correctas	{lot != null}	borra Lote	1		

verificarExistencias()

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Formato tipo	{tipo = TIPXXXXXX && 0 <= X <= 9} 1	
Intervalo	{0 < cantidad <= 999} 2	
Existencia	{tipo pertenece a un TipoProducto existente} 3	

Batería de pruebas

	Entradas	Salidas	Clases cubiertas	Valores Límite	Salidas
Clases correctas	{TIP000001, 300}	No alcanzan las existencias para aprobar el pedido	1,2,3	{1}	Alcanzan las existencias para aprobar el pedido
				{999}	No alcanzan las existencias para aprobar el pedido

getCodigo()

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Codigo	{"Flipper" || "Consola individual" || "Consola grupal" || "Simulador"}1	

Batería de pruebas

	Entradas	Salidas	Clases cubiertas	Valores Límite	Salidas
Clases correctas	{"Flipper"}	TIP000001	1		

getProducto()

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Formato codigo	{codigo = TIPXXXXXX && 0 <= X <= 9} 1	
Existencia	{codigo pertenece a un TipoProducto existente} 2	

Batería de pruebas

	Entradas	Salidas	Clases cubiertas	Valores Límite	Salidas
Clases correctas	{"TIPXXXXXX"}	TipoProducto correspondiente	1,2		

actualizarExistencias()

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Tipo Producto	{tipo != null} 1	

Batería de pruebas

	Entradas	Salidas	Clases cubiertas	Valores Límite	Salidas
Clases correctas	{tipo != null}	el stock resulta modificado	1		

Constructor

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Formato legajo	{legajo = LEGXXXXXX && 0 <= X <= 9} 1	
Longitud nya	{0 < length nya <= 100}2	
Sector	{"Ventas" || "Produccion" || "Contabilidad" || "Inspeccion y calidad"} 3	

Batería de pruebas

	Entradas	Salidas	Clases Cubiertas	Valores Límite	Salidas
Clases correctas	{ LEG213431,"Rodrigo Cassanelli", "Ventas"}	nueva instancia de la clase	1,2,3	{LEG999999}	nueva instancia De la clase
				{LEG000000}	
				{nya.length = 1}	
				{nya.length = 100}	

Constructor

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Formato numeroPedido	{numeroPedido = PEDXXXXXX && 0 <= X <= 9} 1	
Formato codigoMaquina	{codigoMaquina = TIPXXXXXX && 0 <= X <= 9} 3	
Maquina	{"Flipper" || "Consola individual" || "Consola grupal" || "Simulador"}5	
Intervalo	{0 < cantidad <= 999}7	
fechaPedido	{fechaPedido != null}9	
fechaEntregaVentas	{fechaEntregaVentas != null}11	

Batería de pruebas

	Entradas	Salidas	Clases Cubiertas	Valores Límite	Salidas
Clases correctas	{PED102111, fechaPedido != null, FechaEntregaVentas != null, "Flipper", TIP213431, 200}	nueva instancia de la clase	1,3,5,7,9,11	{PED999999}	nueva instancia De la clase
				{PED000000}	
				{TIP999999}	
				{TIP000000}	
				{1}	
				{999}	

agregarObservacion()

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Observacion	{obs != null} 1	
Estado	{pedido.estadoActual = Evaluacion} 2	{pedido.estadoActual != Evaluacion} 3
Tema	{obs.tema != null} 4	
Legajo	{obs.legajoEmpleado = LEGXXXXXX && 0<=X<=9} 5	
Texto	{obs.texto != null} 6	
FechaObservacion	{obs.fechaObservacion != null} 7	

Batería de pruebas

	Entradas	Salidas	Clases cubiertas	Valores Límite	Salidas
Clases correctas	{obs != null, pedido.estadoActual = Evaluacion, obs.tema != "prueba", obs.legajoEmpleado = "LEGXXXXXX", obs.texto != "es una observacion" , obs.fechaObservacion != null }	observación agregada	1,2,4,5,6,7		
Clases erróneas	{pedido.estadoActual != Evaluacion}	Observacion invalida	3		

aceptarPedido()

Tabla de particiones

Condición	Clases correctas	Clases erróneas
fechaProduccion	{fechaProduccion != null} 1	
Estado	{pedido.estadoActual = Evaluacion} 2	

Batería de pruebas

	Entradas	Salidas	Clases cubiertas	Valores Límite	Salidas

Clases correctas	{fechaProduccion != null, pedido.estadoActual = Evaluacion}	pedido aceptado	1,2		

evaluarPedido()

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Estado	{pedido.estadoActual = Iniciado} 1	

Batería de pruebas

	Entradas	Salidas	Clases cubiertas	Valores Límite	Salidas
Clases correctas	{pedido.estadoActual = Evaluacion}	pedido en evaluación	1		

setEstadoActual()

Tabla de particiones

Condición	Clases correctas	Clases erróneas
estadoActual	{estadoActual != null} 1	{estadoActual == null} 2

Batería de pruebas

	Entradas	Salidas	Clases cubiertas	Valores Límite	Salidas
Clases correctas	{estadoActual != null}	estado seteado	1		
Clases erróneas	{estadoActual == null}	Estado nulo	2		

Constructor

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Formato	{codigo = MATXXXXX && 0 <= X <= 9} 1	{codigo != MATXXXXX} 2
Longitud	{0 < length descripcion <= 100} 3	{length descripcion > 100} 4.1 {length descripcion = 0} 4.2
Intervalo	{000,0000 < cantidad <= 999,9999} 5	{cantidad <= 0}6.1 {cantidad > 999,999}6.2

Batería de pruebas

	Entradas	Salidas	Clases Cubiertas	Valores Límite	Salidas
Clases correctas	{MAT10211, descripcion.lenght = 50, 300}	nueva instancia de la clase	1,3,5	{MAT99999}	nueva instancia De la clase
				{MAT00000}	
				{descripcion.length = 100}	
				{descripcion.length = 1}	
				{999,9999}	
				{000,0001}	

setDescripcion()

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Longitud	{0 < length descripcion <= 100} 1	

Batería de pruebas

	Entradas	Salidas	Clases Cubiertas	Valores Límite	Salidas
Clases correctas	{descripcion.lenght = 60}	setea Descripcion	1	{descripcion.length = 100}	nueva instancia De la clase
				{descripcion.length = 1}	

setCantidad()

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Intervalo	{000,0000 < cantidad <= 999,9999} 1	

Batería de pruebas

	Entradas	Salidas	Clases Cubiertas	Valores Límite	Salidas
Clases correctas	{400}	setea Cantidad	1	{999,9999}	nueva instancia De la clase
				{000,0001}	

Clase Lote

Constructor

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Formato pedido	{pedido != null} 1	
Formato numero de lote	{numeroLote = LOTXXXXXX && 0 <= X <= 9} 3	

Batería de pruebas

	Entrada	Salidas	Clases Cubiertas	Valores limite	Salidas
Clases correctas	(pedido != null , LOT123456)	nueva instancia de la clase	1,3	(pedido != null, LOT000000) (pedido != null, LOT999999)	Nueva instancia de la clase

Clase Observacion

Constructor

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Tema	{tema != null} 1	
Fecha de observacion	{fechaObservacion != null} 3	
Formato legajo	{legajoEmpleado = LEGXXXXXX && 0 <= X <= 9} 5	
Formato texto observación	{0 < length texto <= 500} 7	

Batería de pruebas

	Entrada	Salidas	Clases Cubiertas	Valores limite	Salidas
Clases correctas	(tema != null , fechaObservacion != null, LEG123456, texto.length = 40)	nueva instancia de la clase	1,3,5,7	LEG000000 LEG999999 Texto.length = 1 Texto.length = 500	Nueva instancia De la clase

Clase Aceptado

Constructor

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Pedido	{pedido != null} 1	{pedido == null} 2

Batería de pruebas

	Entrada	Salidas	Clases cubiertas	Valores limite	Salidas
Clases correctas	{pedido != null}	nueva instancia de la clase	1		
Clases erróneas	{pedido == null}	el pedido es null	2		

agregarObservacion

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Observación	{obs != null} 1	

Batería de pruebas

	Entrada	Salidas	Clases cubiertas	Valores limite	Salidas
Clases correctas	{obs != null}	Imposible agregar, pedido ya aceptado	1		

Clase ListaEmpleados

Buscar

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Legajo a buscar	{legajo = LEGXXXXXX && 0<= X < 9} 1	{legajo != LEGXXXXXX} 2

Batería de pruebas

	Entrada	Salidas	Clases Cubiertas	Valores limite	Salidas
Clases correctas	(LEG123456)	Referencia al empelado buscado si este existe, aviso de inexistencia en caso contrario	1	LEG000000 LEG999999	Referencia al empelado buscado si este existe, aviso de inexistencia En caso contrario
Clases erróneas	(LEG123) (LEG1234567)	Aviso de inexistencia del empelado buscado ya que no es un legajo valido	2		

Clase ListaMateriales

agregarMaterial()

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Material a agregar	{nuevo != null} 1	

Batería de pruebas

	Entrada	Salidas	Clases cubiertas	Valores limite	Salidas
Clases correctas	{nuevo != null}	Agregar el nuevo material a la lista	1		

getMaterial()

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Codigo a buscar	{codigo = MATXXXXX && 0 <= X <= 9} 1	

Batería de pruebas

	Entrada	Salidas	Clases Cubiertas	Valores limite	Salidas

Clases correctas	(MAT12345)	Devuelve una referencia la material indicado, o un aviso de inexistencia	1	MAT00000 MAT99999	Devuelve una referencia al material indicado, O un aviso de inexistencia

borrarMaterial()

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Codigo del material a borrar	{codigo = MATXXXXX && 0 <= X <= 9} 1	

Batería de pruebas

	Entrada	Salidas	Clases Cubiertas	Valores limite	Salidas
Clases correctas	(MAT12345)	Borrar el material indicado por el codigo si existe, si no avisa de la inexistencia	1	MAT00000 MAT99999	Borrar el material indicado por el codigo si existe, si no avisa de la inexistencia

agregarMaterial()

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Código del material a agregar	{codigo = MATXXXXX && 0 <= X <= 9} 1	

Descripción del nuevo material	{descripcion != null} 3	
Cantidad del nuevo material	{cantidad >= 0} 5.1 {cantidad ε R } 5.2	

Batería de pruebas

	Entrada	Salidas	Clases Cubiertas	Valores limite	Salidas
Clases correctas	(MAT12345, descripcion.length = 50, 45.2)	Agregado el nuevo material	1,3,5.1,5.2		

Clase ListaPedidos

agregarNuevo()

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Pedido a agregar	{nuevo != null} 1	{nuevo == null} 2

Batería de pruebas

	Entrada	Salidas	Clases cubiertas	Valores limite	Salidas
Clases correctas	{nuevo != null}	Agregar el nuevo pedido a la lista	1		
Clases erróneas	{nuevo == null}	no agrega pedido a la lista	2		

borrarPedido()

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Pedido a eliminar	{ped != null} 1	

Batería de pruebas

	Entrada	Salidas	Clases cubiertas	Valores limite	Salidas
Clases correctas	(ped != null)	Borrar el pedido indicado por el codigo si existe, si no avisa de la inexistencia	1		

Clase Controlador

setEmpleadoActual()

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Referencia al empleado actual	{actual != null} 1	

Batería de pruebas

	Entrada	Salidas	Clases cubiertas	Valores limite	Salidas
Clases correctas	(actual != null)	Setea el empleado actual	1		

setPedidoActual()

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Referencia al pedido actual	{pedido != null} 1	

Batería de pruebas

	Entrada	Salidas	Clases cubiertas	Valores limite	Salidas
Clases correctas	(pedido != null)	Setea el pedido actual	1		

setLoteActual()

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Referencia al lote actual	{lote != null} 1	

Batería de pruebas

	Entrada	Salidas	Clases cubiertas	Valores limite	Salidas
Clases correctas	(lote != null)	Setea el lote actual	1		

setProductoActual()

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Referencia al producto actual	{producto != null} 1	

Batería de pruebas

	Entrada	Salidas	Clases cubiertas	Valores limite	Salidas
Clases correctas	(producto != null)	Setea el producto actual	1		

buscarEmpleado()

Tabla de particiones

Condición	Clases correctas	Clases erróneas

Legajo a buscar	{legajo = LEGXXXXXX && 0<= X < 9} 1	{legajo != LEGXXXXXX} 2

Batería de pruebas

	Entrada	Salidas	Clases cubiertas	Valores limite	Salidas
Clases correctas	(LEG123456)	Referencia al empelado buscado si este existe, aviso de inexistencia en caso contrario	1	LEG000000 LEG999999	Referencia al empelado buscado si este existe, aviso de inexistencia en caso contrario
Clases erróneas	(LEG123) (LEG1234567)	Aviso de inexistencia del empleado buscado ya que no es un legajo valido	2		

cambiarAAceptado()

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Fecha propuesta por produccion	{fechaProduccion != null} 1	

Batería de pruebas

	Entrada	Salidas	Clases cubiertas	Valores limite	Salidas

Clases correctas	(fechaProduccion != null)	Acepta el pedido con la fecha propuesta por produccion	1		

crearNuevoPedido()

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Fecha de pedido	{fechaProduccion != null} 1	{fechaProduccion == null} 2
Tipo de maquina	{tipoMaquina == TIPXXXXXX && 0<= X <= 9} 3	{tipoMaquina != TIPXXXXXX} 4
Cantidad a producir	{cantProducir >= 0} 5	{cantProducir < 0} 6
Fecha solicitada por ventas	{fechaVentas != null} 7	{fechaVentas == null} 8

Batería de pruebas

	Entrada	Salidas	Clases cubiertas	Valores limite	Salidas
Clases correctas	(fechaProduccion != null, TIP123456, 200, FechaVentas != null)	Nuevo pedido creado	1,3,5,7	TIP000000 TIP999999	Nuevo pedido creado
Clases erróneas	fechaProduccion == null FechaVentas == null CantProducir = -1 TIP123 TIP1234567	Error	2 8 6 4 4		

crearObservacion()

Tabla de particiones

Condición	Clases correctas	Clases erróneas
Tema	{tema != null} 1	{tema == null} 2
Formato texto observación	{0 < length texto <= 500} 3	{length texto <= 0 } 4.1 {length texto > 500} 4.2

Batería de pruebas

	Entrada	Salidas	Clases cubiertas	Valores limite	Salidas
Clases correctas	(tema != null, texto.length = 50)	Nueva observación agregada	1,3	texto.length = 1 Texto.length = 500	Nueva observación agregada
Clases erróneas	Tema == null Texto.length = 0 Texto.length = 501	Informa del error y no agrega la observación	2 4.1 4.2		