N° Alumno	Diseño de pruebas Comentario Diseño de pruebas Coverage parte A	Comentario Coverage parte A Coverage parte E	Comentario Coverage parte B Pregunta Conceptual 1	Comentario P1	Progunta Concentual 2	Comontario P2	Progunts Concentual 3	Comentario P3	Progunts Concentral 4 Comentario P4 Description	nto Atraeo Nota
N Alumno	Disento de pruebas Comentario Disento de pruebas Coverage parte A	Comentario Coverage parte A Coverage parte E	Contentario Coverage parte B Pregunta Conceptuar i	Comentario Fi	Fregunta Conceptual 2	Se habla de costos	Pregunta Conceptuar 3	Comentario F3	Pregunta Conceptual 4 Comentano P4 Descuer	ILO ALI ASO NOLA
				En una función con	()	generales, la			Las más	
				dominio y output acotado si se		segunda relacionar con madurez de			importantes son de sistemas e	
12638781	1.5 ¡Perfecto! 1	.5	1 Muy bien! 0.:	25 puede		proyecto	0.5	Bien	0.25 integración	6.00
			Faltó el grafo (-0.2). Alumno 4 del							
			conjunto 2 si entra al else if (falla parcial en análisis de branch							
	Hay test que evalúan lo mismo como		coverage, -0.2). El conditional					Falto decir depende	,	
	el segundo y el último que fallan por	Falta 1 test en Branch Coverage.	coverage no evalúa cada condición	En una función con	u l			ya que un % de		
	no tener el largo mínimo. Uno de ellos no aporta más a detectar	(El caso 1 de Condition Coverage deberia ser tambien de Branch	completa del if, sino que cada statement booleano por si solo	dominio y output acotado si se		No buena		coverage no garantiza mejor	Faltó enfoque	
13620916	1.2 problemas. 1.2	25 Coverage)	0.3 (-0.3).	25 puede	0.25	justifiación	0	calidad	0.25 regresión	4.50
13634372										1.00
			En general bien, pero ojo con la							
			notación de grafos (-0.1), si bien se entiende las anotaciones de lineas,	En una función con		El número de				
	Faltó el test donde se espera True		en rigor se deben representar	dominio y output	'	líneas está				
13638386	cuando se cumplen todos los 1.25 requisitos.	5	bloques de código como nodos v	acotado si se 25 puede	0.25	relacionado, pero no es el factor	0.5	Bien	Falta enfoque	5.90
13030300	1.25 requisitos.	.5	Bien en general, pero ojo con la	25 puede	0.23	no es el lactor	0.5	Dieli	0.25 regresión	5.90
			notación del grafo (-0.1). Cada							
			bloque de código es un nodo y cada							
		Falta un test en condition coverage:	condición es una bifuración. En tu caso, solo habrían 3 nodos y 3	En una función con						
		Tiene dinero suficiente, no es	ramas. El análisis de branch	dominio y output	'				Nombras	
14619083	1.5 Perfecto! 1.3	recomendado, es el volumen 1 y	coverage está bien hecho bajo la	acotado si se		Bien		Bien	demasiados en la	
14019063	1.5 [Fellecto: 1.2	25 tiene más de 7 volúmenes totales.	0.9 lógica de tu grafo. 0.:	25 puede	0.8	Dieli	0.5	Didli	0.25 segunda parte	6.15
			¡Bien! Pero ojo con los conceptos para el branch coverage. La rama							
			es de un nodo a otro, lo que					1		
14621150	1.5 Perfecto! 1.2	25	mencionaron en su respuesta son 1 caminos.	0.5 Bien	0.5	Bien	0.5	Bien	0.25 Falta regresión	6.50
1.102.1.00	La función retorna True si cumple				0.0	1	0.0		0.20 i ana regressori	0.50
	todos los requisitos, en otro caso							1		
	retorna False por no ser segura. En cada test agregaron esperar un True									
	por "si no se cae", pero esos test		¡Muy bien! Lo único es que para la					1		
	arrojan False porque no cumplen		otra consideren que los ands							
	todos los requisitos para ser una contraseña segura. Además, para		pueden no evaluarse si la claúsula anterior es False. No está mal, pero							
	cada requisito agregaron otro test		es un supuesto que deberían							
	que efectivamente retorna True si se	Statement Coverage no co minimo	mencionar y que en este caso se	En una función	.1			1		
	cumplen todos los requisitos, pero eso hace que testeen 5 veces el	Statement Coverage no es minimo (Deberian ser 3 tests). Falta un test	infiere de su respuesta. Consideren que sin ese supuesto, el primer	En una función con dominio y output	1					
	requisito de: retornar True si pasa	en Condition Coverage (revisar las	conjunto tiene menos condition	acotado si se		.				
14632527	1.2 todas las condiciones.	1 condicions con OR)	1 coverage. 0	25 puede	0.5	Bien	0.5	Bien	0.5 Bien	5.95
14633426	Unit of the second of the seco	+		+	1	-	-	+		1.00
	Había un test que se podría haber omitido, que es el de más de 8							1		1
	caracteres que retorna True porque			L						
	el primer test ya valida que se cumpla el mínimo (8 o más), pero			En una función con dominio y output	1			1		1
	somo es solo un test, no se aplica		Buen análisis, pero faltó el grafo	acotado si se				1		
14633450	1.5 descuento. 1	.5	0.8 (-0.2)	25 puede	0.5	Bien	0.5	Bien	0.5 Bien	2 4.55
		El test 1 no deberia estar en	¡Muy bien! Eso si, el conditional coverage pide que cada statement							
		Statement Coverage. Hay 2 tests	booleano por si solo evalue a true y	En una función con				1		
	Faltó el test donde se espera True	que son extra en Condition	false, no en conjunto. No se	dominio v output						
14636530	cuando se cumplen todos los 1.25 requisitos.	Coverage, deberia ser el minimo (6 1 tests)	descuenta porque la fuente puede 1 ser válida igual. 0.:	acotado si se 25 puede	0.5	Bien	0.5	Bien	Enfoque de 0.25 regresión	5.75
		,	0.	En una función con		1	0.3		regression	5.75
				dominio y output				1		
15204030	1.5 iPerfecto!	0 No entrega pregunta	0 No entrega pregunta 0.:	acotado si se	0.5	Bien	0.5	Bien	Falta enfoque 0.25 regresión	4.00
13204030	1.5 Gliccio:	o no omoga preguna	U.s	En una funció con	0.3	, , , , ,	0.5	Dioi!	0.20 Tegresion	4.00
		No separa Condition y Branch		dominio y ouput				1	Mal enfoque y	
15621146	1.5 iPerfecto!	coverage. Se buscan los tests 1 minimos para cada uno	1 iMuv bien! 0.	acotado si se	1	Bien		Bien	luego nivel	5.75
15021146	1.5 ¡Periecto! Dada la justificación entregada,	i minimos para cada uno	i jiwiuy bieti: 0.3	20 puede	0.8	pieri	0.5	Dieil	U tradicional	5.75
	algunos test no eran necesarios, en							1		
		1		1	1	1	1	1		
	otras palabras, el beneficio de							1		
	otras palabras, el beneficio de algunos test entregados ya es suplido por otros. Para que la función									
	algunos test entregados ya es suplido por otros. Para que la función retorne True, se deben cumplir todos									
	algunos test entregados ya es suplido por otros. Para que la función retorne True, se deben cumplir todos los requisitos. Probar en cada uno									
	algunos test entregados ya es suplido por otros. Para que la función retorne True, se deben cumplir todos los requisitos. Probar en cada uno que se retorne True es similar a hacer un test oue retorne True (dado	Branch Coverage y Condition		En una función con				Se debe decir que		
	algunos test entregados ya es supido por otros. Para que la función retorne True, se deben cumplir todos los requisitos. Proba en cada uno testa de la companio de la companio de la hacer un test que retorne True (dado que seu n.ADI). Falto, tal vez, incluir	Branch Coverage y Condition Coverage no tiernen el numero		dominio y output	r			depende, ya que el		
15621219	algunos test entregados ya es suplido por otros. Para que la función retorne True, se deben cumplir todos los requisitos. Probar en cada uno que se retorne True es similar a hacer un test que retorne True (dado que es un AND). Fallo, tal vez, incluir una justificación de que probarás	Branch Coverage y Condition Coverage no listen of humero minimo de tests. Deberian ser 4 y 6 1 respectivamente	1 Muy bient 0.1			5 Bien	0	depende, ya que el % no garantiza	0.5 Bien	5.45
15621219	algunos test entregados ya es supido por otros. Para que la función retorne True, se deben cumplir todos los requisitos. Proba en cada uno testa de la companio de la companio de la hacer un test que retorne True (dado que seu n.ADI). Falto, tal vez, incluir	minimo de tests. Deberian ser 4 y 6 1 respectivamente	1 ¡Muy bien! 0.:	dominio y output acotado si se	0.0	5 Bien	0	depende, ya que el	0.5 Bien	5.48
15621219	algunos test entregados ya es suplido por otros. Para que la función retorne True, se deben cumplir todos los requisitos. Probar en cada uno que se retorne True es similar a que se un AND). Falló. La tez, incluir una justificación de que probarás 1.2 casos bordes para llegar al True.	minimo de tests. Deberian ser 4 y 6 1 respectivamente El numero de tests en Branch y	1 jMuy bien! 0.:	dominio y output acotado si se 25 puede En una función con dominio y output	0.0	5 Bien	0	depende, ya que el % no garantiza	Falto decir	5.48
	algunos test entregados ya es suplido por otros. Para que la función retorne True, se deben cumplir todos los requisitos. Probar en cada uno que se retorne True es similar a hacer un test que retorne True (dado que es un AND). Fallo, tal vez, incluir una justificación de que probarás	minimo de tests. Deberian ser 4 y 6 1 respectivamente		dominio y output acotado si se 25 puede En una función con dominio y output acotado si se	0.6	5 Bien		depende, ya que el % no garantiza	Falto decir explicitamente	
15621219 15621278	algunos test entregados ya es suplido por otros. Para que la función retorne True, se deben cumplir todos los requisitos. Probar en cada uno que se retorne True es similar a hacer un test que retorne True (dado que es un AND). Falto, tal vez, incluir una justificación de que probarás 1.2 casos bordes para llegar al True. Falta el test donde falte letra	minimo de tests. Deberian ser 4 y 6 1 respectivamente El numero de tests en Branch y Condition Coverage no son		dominio y output acotado si se 25 puede En una función con dominio y output	2.0			depende, ya que el % no garantiza mejor codigo	Falto decir	5.45
	algunos test entregados ya es suplido por otros. Para que la función retorne True, se deben cumplir todos los requisitos. Probar en cada uno que se retorne True es similar a hacer un test que retorne True (dado que es un AND). Falto, tal vez, incluir una justificación de que probarás 1.2 casos bordes para llegar al True. Falta el test donde falte letra	minimo de tests. Deberian ser 4 y 6 Tespectivamente El numero de tests en Branch y Condition Coverage no son 1 minimos. No separa Condition y Branch		dominio y output acotado si se 25 puede En una función con dominio y output acotado si se 25 puede En una función con dominio y ouput	2.0			depende, ya que el % no garantiza mejor codigo	Falto decir explicitamente 0.25 regresión	
15621278	algunos test entregados ya es supido por otros. Para que la función retorne True, se deben cumplir todos que ser tente en entre a hacer un test que retorne True es similar a hacer un test que retorne True (dado que es un AND). Falto, la text, incluir una justificación de que probarás 1.2 casos bordes para llegar al True. Falta el test donde falte letra minúscula.	minimo de tests. Deberian ser 4 y 6 I respectivament El numero de tests en Branch y Condition Coverage no son minimos. No separa Condition y Branch coverage. Se buscan los tests	1 Bien 0.:	dominio y output acotado si se 25 puede En una función con dominio y output acotado si se 25 puede En una función con dominio y ouput acotado si se	0.5	5 Bien	0.5	depende, ya que el % no garantiza mejor codigo Bien	Falto decir explicitamente 0.25 regresión Mal enfoque y luego nivel	5.75
	algunos test entregados ya es suplido por otros. Para que la función retorne True, se deben cumplir todos los requisitos. Probar en cada uno que se retorne True es similar a hacer un test que retorne True (dado que es un AND). Falto, tal vez, incluir una justificación de que probarás 1.2 casos bordes para llegar al True. Falta el test donde falte letra	minimo de tests. Deberian ser 4 y 6 Tespectivamente El numero de tests en Branch y Condition Coverage no son 1 minimos. No separa Condition y Branch	1 Bien 0.1	dominio y output acotado si se 25 puede En una función con dominio y output acotado si se 25 puede En una función con dominio y ouput	0.5		0.5	depende, ya que el % no garantiza mejor codigo	Falto decir explicitamente 0.25 regresión	
15621278	algunos test entregados ya es supido por otros. Para que la función retorne True, se deben cumplir todos que ser tente en entre a hacer un test que retorne True es similar a hacer un test que retorne True (dado que es un AND). Falto, la text, incluir una justificación de que probarás 1.2 casos bordes para llegar al True. Falta el test donde falte letra minúscula.	minimo de tests. Deberian ser 4 y 6 I respectivament El numero de tests en Branch y Condition Coverage no son minimos. No separa Condition y Branch coverage. Se buscan los tests	1 Muy blen! 0.: Fatto el grafo (-0.2), La idea del branch coverçace es que se analiza	dominio y output acotado si se 25 puede En una función con dominio y output acotado si se 25 puede En una función con dominio y ouput acotado si se	0.5	5 Bien	0.5	depende, ya que el % no garantiza mejor codigo Bien	Falto decir explicitamente 0.25 regresión Mal enfoque y luego nivel	5.75
15621278	algunos test entregados ya es supido por otros. Para que la función retorne True, se deben cumplir todos que ser tente en entre a hacer un test que retorne True es similar a hacer un test que retorne True (dado que es un AND). Falto, la text, incluir una justificación de que probarás 1.2 casos bordes para llegar al True. Falta el test donde falte letra minúscula.	minimo de tests. Deberian ser 4 y 6 I respectivament El numero de tests en Branch y Condition Coverage no son minimos. No separa Condition y Branch coverage. Se buscan los tests	1 Bien 0.: 1 Muy bien! 0.: Falló el grafo (-0.2). La idea del branch coverage es que se analiza por las bifurçaciones posibles (P y	dominio y output acotado si se 25 puede En una función con dominio y output acotado si se 25 puede En una función con dominio y ouput acotado si se	0.5	5 Bien	0.5	depende, ya que el % no garantiza mejor codigo Bien	Falto decir explicitamente 0.25 regresión Mal enfoque y luego nivel	5.75
15621278	algunos test entregados ya es supido por otros. Para que la función retorne True, se deben cumplir todos que ser tente en entre a hacer un test que retorne True es similar a hacer un test que retorne True (dado que es un AND). Falto, la text, incluir una justificación de que probarás 1.2 casos bordes para llegar al True. Falta el test donde falte letra minúscula.	minimo de tests. Deberian ser 4 y 6 I respectivament El numero de tests en Branch y Condition Coverage no son minimos. No separa Condition y Branch coverage. Se buscan los tests	1 Muy blen! 0.: Fatto et grafo (-0.2), La idea del branch coverage es que se analiza por las bifuracaciones posibles (P y T >= 4.0 o T >= 4.5 P >= 3,8 y A >=	dominio y output acotado si se 25 puede En una función con dominio y output acotado si se 25 puede En una función con dominio y ouput acotado si se	0.5	5 Bien	0.5	depende, ya que el % no garantiza mejor codigo Bien	Falto decir explicitamente 0.25 regresión Mal enfoque y luego nivel	5.75
15621278	algunos test entregados ya es supido por otros. Para que la función retorne True, se deben cumplir todos que ser tente en entre a hacer un test que retorne True es similar a hacer un test que retorne True (dado que es un AND). Falto, la text, incluir una justificación de que probarás 1.2 casos bordes para llegar al True. Falta el test donde falte letra minúscula.	minimo de tests. Deberían ser 4 y 6 I respectivamente El numero de tests en Branch y Condition Coverage no son I minimos. No separa Condition y Branch coverage. Se buscan los tests I minimos para cada uno.	1 Muy bien! 0. Faito el grafo (-0.2). La idea del branch coverage es que se analiza branch coverage es que se analiza branch coverage es que se analiza proposition (-0.01 ≥ -4.5, P >= 3.8 s (P) = -0.9), por lo que esos casos que mencionan use no se cubren no	dominio y output acotado si se puede En una función con dominio y output acotado si se puede En una función con dominio y output acotado si se puede En una función con dominio y output acotado si se puede En una función con	0.1	5 Bien	0.5	depende, ya que el % no garantiza mejor codigo Bien	Falto decir explicitamente 0.25 regresión Mal enfoque y luego nivel	5.75
15621278	algunos test entregados ya es supido por otros. Para que la función retorne True, se deben cumplir todos que ser tente en entre a hacer un test que retorne True es similar a hacer un test que retorne True (dado que es un AND). Falto, la text, incluir una justificación de que probarás 1.2 casos bordes para llegar al True. Falta el test donde falte letra minúscula.	minimo de tests. Deberian ser 4 y 6 Tespectivamente El numero de tests en Branch y Condition Coverage no son 1 minimos. No separa Condition y Branch coverage. Se buscan los tests 1 minimos para cada uno.	Bien 0.:	dominio y output acotado si se 25 puede En una función con dominio y output acotado si se 25 puede En una función con dominio y output acotado si se En una función con dominio y ouput acotado si se puede En una función con dominio y output acotado si se	0.1	5 Bien	0.5	depende, ya que el % no garantiza mejor codigo Bien	Falto decir explicitamente 0.25 regresión Mal enfoque y luego nivel	5.75
15621278	algunos test entregados ya es supido por otros. Para que la función retorne True, se deben cumplir todos que ser tente en entre a hacer un test que retorne True es similar a hacer un test que retorne True (dado que es un AND). Falto, la text, incluir una justificación de que probarás 1.2 casos bordes para llegar al True. Falta el test donde falte letra minúscula.	minimo de tests. Deberían ser 4 y 6 I respectivamente El numero de tests en Branch y Condition Coverage no son I minimos. No separa Condition y Branch coverage. Se buscan los tests I minimos para cada uno.	1 Muy bien! 0. Fatio el grato (-0.2). La idea del branch coverage es que se analiza branch coverage es que se analiza P Y 2 = 4,0 o T ≥ 4,5 p ≥ 3,8 y Å ≥ 0,9), por lo que esos casos que mencionan que no se cubren no aplican para el análisis (falla completamente en análisis del	dominio y output acotado si se puede En una función con dominio y output acotado si se puede En una función con dominio y output acotado si se puede En una función con dominio y output acotado si se puede En una función con	2.0	5 Bien	0.5	depende, ya que el % no garantiza mejor codigo Bien	Falto decir explicitamente 0.25 regresión Mal enfoque y luego nivel	5.75

	Dada la justificación entregada,							
	algunos test no eran necesarios, en otras palabras, el beneficio de							
	algunos test entregados va es							
	suplido por otros. Para que la función							
	retorne True, se deben cumplir todos							
	los requisitos. Probar en cada uno que se retorne True es similar a							
	hacer un test que retorne True (dado		Buen análisis de branch coverage,					
	que es un AND). Faltó, tal vez, incluir	Falta 1 test en Branch Coverage.	pero al de condition le faltan					
	una justificación de que probarás	Hay 1 test extra en Condition	números exactos de casos posibles					
15623688	1.2 casos bordes para llegar al True.	1 Coverage	0.8 y de casos logrados (-0.2)	0 No contesta	0 No contesta	0 No contesta	0 No contesta	4.00
		El numero de tests en Branch y		En una función con dominio y output			Falto decir	
	Falta el test donde falte letra	Condition Coverage no son		acotado si se			explicitamente	
15624005	1.25 minúscula.	1 minimos.	1 Bien	0.25 puede	0.5 Bien	0.5 Bien	0.25 regresión	5.75
			Bien en general, pero si bien se					
			entiende que N3 representa a aprobado = True, en rigor no deben					
			ser el mismo nodo (son bloques					
			distintos de código, -0.1). Respecto					
			al conditional coverage, ojo para el futuro que pide que cada statement	En una función con				
			booleano sea evaluado como true o	dominio y output			Falto decir	
		Hay un test extra en Condition	false por si solo, no en base al	acotado si se			explicitamente	
15633144	1.5 ¡Perfecto!	1.25 Coverage. Deberian ser 6	0.9 predicado completo.	0.25 puede	0.5 Bien	0.5 Bien	0.25 regresión	6.15
				En una función con dominio y output	Determina la			
		Faltan 2 tests en Condition	Buen análisis, pero faltó el grafo	acotado si se	madurez del		Fata de sistema o	
15633241	1.5 ¡Perfecto!	1.25 Coverage	0.8 (-0.2).	0.25 puede	0 proyecto	0.5 Bien	0.25 integracion	5.55
			¡Sobresaliente! Lo único es que					
			para la otra consideren que los ands pueden no evaluarse si la					
			claúsula anterior (n. n. r. s. o.t) es					
			claúsula anterior (p, q, r, s o t) es False. No está mal, pero es un					
			supuesto que deberían mencionar y que en este caso se infiere de sus				Falta nombrar	
			que en este caso se infiere de sus tablas bien presentadas :).	En una función con		Un % de coverage	Falta nombrar regresión y hay	
		Hay un test extra en Condition	Consideren que sin ese supuesto,	dominio y output		no garantiza un	algunos más	
		Coverage, Deberian ser 6 tests	el primer conjunto tiene menos	acotado si se		meior calidad de	importantes que	
15633268	1.5 ¡Perfecto!	1.25 para esa cobertura.	1 condition coverage.	0.25 puede	0.5 Bien	0.25 código	0 otros	5.75
			Buen análisis en general, pero ojo que el conditional coverage pide					
			que cada statement booleano de					
			manera independiente sea true y	En una función con				
		Falta 1 test en Branch Coverage.	false, lo que cambia los casos posibles. Al no ser materia oficial,	dominio y output acotado si se				
15634108	1.5 iPerfecto!	1.25 (Test ID=4)	1 no hay penalización.	0.25 puede	0.5 Bien	0.5 Bien	0.5 Bien	6.50
		,	No se entiende muy bien sus					
			resultados para condition coverage:					
			en el conjunto 1 alumno 2 (asumo					
			que el segundo bullet point), a pesar de que A && B == False, no					
			siguieron con los valores booleanos					
	Solo era necesario 1 test por cada		de las condiciones C, D y F que					
	criterio que diga False y 1 test que diga True si pasa todos los test. Hay		eran la rama que seguía, entonces omitieron ciertos casos y sus					
	test redundantes cuya justificación es	Faltaron tests de las condiciones	lanálisis de condition coverage no					
15634809	1.2 la misma.	1 con "OR" en condition coverage.	0.7 están 100% correctos. (-0.3)	0.5 Bien	0.5 Bien	0.5 Bien	0.25 Falta regresión	5.65
			No se entiende muy bien sus					
			resultados para condition coverage: en el conjunto 1 alumno 2 (asumo					
			que el segundo bullet point), a					
			pesar de que A && B == False, no					
	Solo era necesario 1 test por cada		siguieron con los valores booleanos de las condiciones C, D y F que					
	criterio que diga False y 1 test que		eran la rama que seguía, entonces					
	diga True si pasa todos los test. Hay		omitieron ciertos casos y sus					
15634884	test redundantes cuya justificación es 1.2 la misma.	Faltaron tests de las condiciones	análisis de condition coverage no	0.5 Bien	0.5 Bien	0.5 Bien	0.05 5-8	5.05
13034004	1.2 ld IIIISIIId.	1 con "OR" en condition coverage.	0.7 están 100% correctos. (-0.3) ¡Sobresaliente! Lo único es que	U.S Biell	U.S Biell	U.S Biell	0.25 Falta regresión	5.65
			para la otra consideren que los					
1 1			ands pueden no evaluarse si la					
			claúsula anterior (p, q, r, s o t) es					
			False. No está mal, pero es un supuesto que deberían mencionar y					
			que en este caso se infiere de sus				Falta nombrar	
		Use and the second seco	tablas bien presentadas :). Consideren que sin ese supuesto,	En una función con		Un % de coverage	regresión y hay	
		Hay un test extra en Condition Coverage. Deberian ser 6 tests	Consideren que sin ese supuesto, el primer conjunto tiene menos	dominio y output acotado si se		no garantiza un mejor calidad de	algunos más importantes que	
15635058	1.5 ¡Perfecto!	1.25 para esa cobertura.	1 condition coverage.	0.25 puede	0.5 Bien	0.25 código	0 otros	5.75
	i i		Buen análisis en base a su grafo,			- J		
			pero cada nodo debiera representar					
			un bloque de código y cada arista una bifurcación. En su caso,					
			utilizaron cada condición como		Si se tiene bajo			
	Agregaron un test de que string vacío		nodo (mal modelamiento del grafo,	En una función con	presupueto, pero el	La respuesta es		
	era True, eso no cumple con las		-0.1). Esto hizo que su branch	dominio y output	proyecto es largo de todas maneras	que un % de	No nombra	
15635198	condiciones dadas por lo que es 1.4 False. Todo lo demás está bien.	1.25 Falta 1 test de Condition Coverage.	coverage fuera igual a su 0.9 conditional coverage.	acotado si se 0.25 puede	0.25 es más "barato"	coverage no 0 garantiza calidad	0.25 regresión	5.30
			Bien en general, pero ojo que			<u> </u>		
	Habían algunos test que evaluaban		condition coverage significa que					
	los mismo (como el de "seña" y		cada condición booleana es al					
	"contra") o verificar con 1, 2, 3 y 4 dígitos. Dada la explicación, era		menos evaluada una vez como true y false, no es necesario que se					
	suficiente con 4 o más porque habían	Falta un test en condition coverage:	considere un conjunto de estas.			Un mayor % de		
	otros test que ya poseían menos de 4 dígitos. Esto hace que se tengan test	Tiene dinero suficiente , no es un	Faltó el grafo (-0.2, a pesar de que		No se habla de la	coverage no garantiza mayor	No so nambro	
15635465	digitos. Esto hace que se tengan test 1.2 redundantes.	libro recomendado y si es el último 1.25 volumen	fuera simple, es necesario ver si 0.8 pueden plasmar la representación)	0.5 Bien	madurez del 0 proyecto	garantiza mayor 0 calidad	No se nombra 0.25 regresión	5.00
15635635			passas passas in reprocentations					1.00
7000000		No separa los tests minimos para						7.00
		cada tipo de Coverage. No estan						
		bien los tests en global (Hay uno	A - 411-1-	₅ ,				
		extra). Deberia estar organizado en	Análisis correcto, pero no hay	En una función con dominio y output		Un porcentaje		
		coverage" "tests 1,2,3,4,5 para	pseudo-código (-0.2). Se infiere que es un es un if/elseif con returns	acotado si se		mayor no garantiza		
15636143	1.5 ¡Perfecto!	1 Branch" y asi	0.8 directos.	0.25 puede	0.5 Bien	0 una mayor calidad	0.25 De regresión	5.30
								

15636585	Una contraseña segura es aquela que cumple todo iso requisitos, por lo tanto tienes test que esperan True cuando será False la respuesta. Por otro lado, tienes test que indician testear funcionalidades distintas pero es el mismo input (qwertyui). Dado esto, solo bastaba hacer 1 de ellos. El test de "QWERTYU" dices que es False porque no posee miniscula y si lo posee. Finalmente, se presentan muchos test que se invalidan por más de una restricción que no se justifica. Se entriega prunisja no se justifica. Se entriega prunisja no se si test, no se alatría exactamente el origen de la falla.	Falta un test en condition coverage: Tiene dinero suficiente, no es recomendado, es el volumen 1 y no 1.25 tiene más de 07 s No separa los tests mínimos para cada tipo de Coverage. Estan bien	Bien en general. Pero ojo que conditional coverage significa que cada condición evalua true y fatse al mence una vez por si sola. Dado que no es materia que hayamos passeo, corregi en base a su suposición.	Puede pasar un 0 test por suerte	No se habla de la madurez del 0 proyecto	0.5 Bien	Enfoque de 0.25 regresión	5.0
15637484	1.5 Perfecto!	los tests en global pero no me dice "test 1,2,3,4 para Statement coverage" "tests 1,2,3,4,5 para 1 Branch" y asi	1 jMuy bien!	En una función con dominio y output acotado si se 0.25 puede	Primera parte hablas de costo, 0.25 demasiado general	Un mayor % de coverage no garantiza mayor 0 calidad	No nombra 0.25 regresión	5.2
15637638	Dada la justificación enfregade, ajujuros test no eran necesarios, en otras palabras, el heneficio de ajujuros test no eran necesarios, en otras palabras, el heneficio de ajujuros test entregados ya es supilido por otros. Para que la función retorme True, se deben cumplir todos los requisitos. Probar en cada uno que ser entome True es similar a hacer un test que retorme True (dado que es un AND). Falto, tal vez, incluir una justificación de que probarás 12 casos bordes para llegar al True.	No separa Condition y Branch coverage. Se buscan los tests 1 minimos para cada uno	1 Muy bient	En una función con dominio y output acotado si se 0.25 puede	Se habla de costo 0.25 general	Un % de coverage no garantiza un 0 mejor código	Enfoque de 0.25 regresión	4.9
15638634	Agregaron un test de que string vacio era True, eso no cumple con las condiciones dadas ano fo nue es	1.25 Falta 1 test de Condition Coverage.	Buen análisis en base a su grafo, pero cada nod debiera representar un bloque de código y cada aristía una blurcación. En su caso, utilizaron cada condición como nodo (mal modelamiento del grafo, 0.1). Esto hizo que su branch coverage fuerá igual a su 0.9 condidional coverage.	En una función con dominio y output acotado si se 0.25 puede	Si se tiene bajo presupueto, pero el proyecto es largo de todas maneras 0.25 es más 'barato'	La respuesta es que un % de coverage no 0 garantiza calidad	No nombra 0.25 regresión	5.3
15638936	1.5 Perfecto!	Habia que separa los tests para cada tipo de coverage y obtener los 0.5 tests minimos para cada uno	Buen análisis de branch coverage, pero al de condition le faltan números exactos de casos posibles 0.8 y de casos logrados (-0.2)	A menos que sea un función con diminio y output acotados no se 0 puede	0.5 Bien	0.5 Bien	Falto enfoque 0.25 regresión	2 3.0
15638944	Una contraseña segura es aquella que cumple todos los requisitos, por lo tanto tienes test que esperan True cuando será Faise la respuesta. Por otro lado, tienes test que indican testear funcionalidades distintas pero es el mismo input (qwertyui). Dado esto, solo bastaba hacer 1 de ellos. El test de "QWERTYU" dices que es Faise porque no posee ministosia y si lo posee. Finalmente, se presentan muchos test que se invalidan por más de una restricción que no se completo proque en caso de fallar tus test test, no se sabría exactamente el oricen de la falla.	Falta un test en condition coverage: Tiene dinero suficiente, no es recomendado, es el volumen 1 y no 1:25 fene más de 7	Bien en general. Pero ojo que conditional coverage significa que cada condición evalua true y false cada condición evalua true y false que no es materia que hayamos pasado, corregí en base a su 1 suposición.	Puede pasar un 0 test por suerte	No se habia de la madurez del 0 provecto	0.5 Blen	Enfoque de 0.25 regresión	5.01
15638952	La función retorna True si cumple todos los requisitos, en otro caso retorna False por no ser segura. En cada test agregaron esperar un True por "sín os e cae", pero seos test en cada test agregaron esperar un True por "sín os e cae", pero seos test en contración de contración de contración de contración de contración acquira. Además, para cada requisito agregaron otro test que efectivamente retorna True si se cumplen todos los requisitos, pero eso hace que testeno "veces el requisito de: retorna True si pasa 12 lodas las condiciones.	Statement Coverage no es minimo (Deberian ser 3 tests). Falta un test en Condition Coverage (revisar las 1 condicions con CR)	I supusación. I Muy bient Lo único es que para la otra consideren que los ands otras consideren que los ands otras consideren que los ands anterior es Eslas. No está mal, pro es un supuesto que deberían mencionar y que en este caso se infere de su respuesta. Consideren que sin ese supuesto, el primer conjunto tiene menos condition 1 coverage.	En una función con dominio y output acotado si se 0.25 puede	0 plyseca	0.5 Blen	0.5 Blen	5.9
15639053		No separa los tests en cada tipo y no son minimos de Condition 0.5 Coverage	Buen anállisis en base a su grafo, pero cada nodo debiera representar un bloque de código y cada arista una bifurcación. En su caso, utilizaron cada condición como nodo (mal modelamiento del grafo, 0.9 0.1).	0.5 Bien	No se habla de la madurez del 0 proyecto	0.5 Bien	Enfoque de regresión y test de sistemas e 0 integración	4.9
15639495	1.5 ¡Perfecto!	Falta 1 test en Branch Coverage y los tests en Condition Coverage no 1 son los minimos (deberian ser 6)	Buen análisis en general, pero faltó rigor en los casos del conditional 0.8 (coverage (-0.2)	En una función con dominio y output acotado si se 0.25 puede	No se contesta 0.25 primera pregunta	Un mejor coverage no determina una 0 mejor calidad	0 No contesta	2 2.8
15639746	1.5 Perfecto!	Falta 1 test en Branch Coverage. 1.25 (Test ID-4)	Buen análisis en general, pero ojo que el conditional coverage pide que cada statement booleano de manera independiente sea true y false, lo que cambia los casos posibles. Al no ser materia oficial, no hay penalización.	En una función con dominio y output acotado si se 0.25 puede	0.5 Bien	0.5 Bien	0.5 Bien	6.51
16203615	Dada la justificación entregada, algunos test no eran necesarios, en otras palabras, el beneficio de algunos test entregados ya es supilido por otros. Para que la función retorme True, se deben cumpilir todos los requisitos. Probar en cada uno que se retorne True es similar a hacer un test que retorne True (dado que es un AND). Facto, tal vez, incluir una justificación de que probarás 12 casos bordes para llegar al True.	Condition Coverage no tiene los 1.25 lests minimos. (Hay 2 tests extras)	Buen análisis, pero faltó el grafo 0.8 (-0.2).	0.5 Bien	No se habia de la madurez del 0, proyecto	depende, % de coverage no g garantiza calidad	Falta enfoque y no se especifica niveles 0 tradicionales	4.7

				En una función con dominio y output				
				acotado si se				
16206851	1.5 ¡Perfecto!	1.5	1 jMuy bien!	0.25 puede	0.5 Bien	0.5 Bien	0.25 De regresión	6.50
			Buen análisis y presentación de los casos, pero se pedía justificación					
			en base al grafo que no aparece					
			(-0.2). Ojo que el análisis de condition coverage no es 100%					
			correcto, pues no se busca evaluar					
			todas las posibles combinaciones		Se habla de tiempo			
			de cada cláusula, sino que todas sean al menos una vez true/false.		ejecución y no de madurez de		No está bien	
			No hay descuento pues no se les		proyecto en la		enfocada la	
16207041	1.5 ¡Perfecto!	1.5	0.8 pasó como materia oficial.	0.5 Bien	0 segunda parte	0.5 Bien	0 respuesta	5.80
			En general buen análisis, pero te equivocaste en el alumno 5 del					
		Statement Coverage no es minimo (Deberian ser 3 tests). Falta un test	conjunto 2, que va por el camino p3 y no p4. Esto hace que el branch					
		(Deberian ser 3 tests). Falta un test	y no p4. Esto hace que el branch		No se habla de la madurez del			
16635019	1.5 ¡Perfecto!	en Condition Coverage (revisar las 1 condicions con OR)	coverage sea 100% y no 75% 0.9 (-0.1).	0.5 Bien	0 proyecto	0.5 Bien	0.5 Bien	5.90
	ĺ	No separa los tests minimos para cada tipo de Coverage. Estan bien						
		cada tipo de Coverage. Estan bien		Farma finally and				
		los tests en global pero no me dice "test 1,2,3,4 para Statement		En una función con dominio y output				
		coverage" "tests 1,2,3,4,5 para		acotado si se	No se contestas la			
16635191	1.5 ¡Perfecto!	1 Branch" y asi	1 ¡Muy bien!	0.25 podría	0.25 segunda pregunta	0.5 Bien	0.5 Bien	6.00
16635221	1.5 ¡Perfecto!	15	1 ¡Muy bien!	0.5 Bien	0.5 Bien	0.5 Bien	Falta enfoque 0.25 regresión	6.75
10000221	1.5 p choos.	Revisar las definiciones de cada	i jindy bioli.	0.0 5.01	0.0 5.01	0.0 51011	U.EU TOGICOION	0.70
		tipo de coverage. Hay un tests extra en condition coverage, deberia ser						
		en condition coverage, deberia ser el minimo numero de pruebas. No	¡Bien en general! Pero ojo con la					
		pueden haber mas tests minimos	notación del grafo (-0.1), las					
		en statement coverage que en	condiciones se evalúan como un único nodo a pesar de tener				E1	
		branch coverage. Uno es "subconjunto" de otro.(Statement	múltiples statements booleanos,				El enfoque más importante es de	
16635264	1.5 ¡Perfecto!	0.75 <= Branch)	0.9 que simplifica bastante el análisis.	0.5 Bien	0.5 Bien	0.5 Bien	0.25 regresión	5.90
	Dada la justificación entregada,							
	algunos test no eran necesarios, en otras palabras, el beneficio de							
	algunos test entregados ya es							
	suplido por otros. Para que la función							
	retorne True, se deben cumplir todos los requisitos. Probar en cada uno							
	que se retorne True es similar a							
	hacer un test que retorne True (dado	No concre Condition y Propeh		En una función con		Un % de coverage		
	que es un AND). Faltó, tal vez, incluir una justificación de que probarás	No separa Condition y Branch coverage. Se buscan los tests		dominio y output acotado si se	Se habla de costo	no garantiza un	Enfoque de	
16635345	1.2 casos bordes para llegar al True.	1 minimos para cada uno	1 Muy bien!	0.25 puede	0.25 general	0 mejor código	0.25 regresión	4.95
16635361					3	3,		1.00
16635361			Bien el análisis de condition		1.00 garate	3, 3, 3, 3		1.00
16635361			coverage, pero faltó todo lo de			7		1.00
16635361	, 5		coverage, pero faltó todo lo de branch coverage (-0.3). También, ojo con la notación del grafo (-0.1).		3			1.00
16635361			coverage, pero faltó todo lo de branch coverage (-0.3). También, ojo con la notación del grafo (-0.1). Cada bloque de código es un nodo,	En una función con		Falta explicitar que	Falls correction as	1.00
		Falta un test en Branch Coverage y	coverage, pero falló todo lo de branch coverage (-0.3). También, ojo con la notación del grafo (-0.1). Cada bloque de código es un nodo, y los controles de fluio definen las	En una función con dominio y output acotado si se		Falta explicitar que el % de coverace	Falló regresión y se nombraron	
16635361 16635442	1.5 ¡Perfecto!	Falta un test en Branch Coverage y 1 en Condition Coverage	coverage, pero faltó todo lo de branch coverage (-0.3). También, ojo con la notación del grafo (-0.1). Cada bloque de código es un nodo,	En una función con	0.5 Bien	Falta explicitar que	Faltó regresión y se nombraron 0 demasiados	4.85
		No separa los tests minimos para	coverage, pero falló todo lo de branch coverage (-0.3). También, ojo con la notación del grafo (-0.1). Cada bloque de código es un nodo, y los controles de fluio definen las	En una función con dominio y output acotado si se		Falta explicitar que el % de coverace	nombraron	
		No separa los tests minimos para	coverage, pero falló todo lo de branch coverage (-0.3). También, ojo con la notación del grafo (-0.1). Cada bloque de código es un nodo, y los controles de fluio definen las	En una función con dominio y output acotado si se		Falta explicitar que el % de coverace	nombraron	
		No separa los tests minimos para cada tipo de Coverage. Estan bien los tests en global pero no me dice "test 1 2 3 4 para Statement	coverage, pero falló todo lo de branch coverage (-0.3). También, ojo con la notación del grafo (-0.1). Cada bloque de código es un nodo, y los controles de fluio definen las	En una función con dominio y output accotado si se 0.25 puede En una función con dominio y output	0.5 Bien	Falta explicitar que el % de coverace	nombraron	
16635442	1.5 ¡Perfecto!	No separa los tests minimos para cada tipo de Coverage. Estan bien los tests en global pero no me dice "test 1 2 3 4 para Statement	coverage, pero falló todo lo de branch coverage, (0.3). También, ojo con la notación del grafo (0.1). Cada bloque de odigo es un nodo, y los controles de flujo definen las ramas, no se debe analizar cada 0.6 condición por separado.	En una función con dominio y output acotado si se 0.25 puede En una función con dominio y output acotado si se	0.5 Bien No se contestas la	Falta explicitar que el % de coverage no garantiza 0 calidad	nombraron 0 demasiados	4.85
16635442 16635469		No separa los tests minimos para	coverage, pero falló todo lo de branch coverage (-0.3). También, ojo con la notación del grafo (-0.1). Cada bloque de código es un nodo, y los controles de fluio definen las	En una función con dominio y output accotado si se 0.25 puede En una función con dominio y output	0.5 Bien	Falta explicitar que el % de coverace	nombraron	4.85
16635442	1.5 ¡Perfecto!	No separa los tests minimos para cada tipo de Coverage. Estan bien los tests en global pero no me dice "test 1 2 3 4 para Statement	coverage, pero falló todo lo de branch coverage, (c.3.). También, ojo con la notación del grafo (c.1.). Cada bloque de oddigo es un nodo, y los controles de flujo definen las manso, no se debe analizar cada controles de flujo definen las comas, no se debe analizar cada controles por separado. 1 jiMuy bien!	En una función con dominio y output acotado si se 0.25 puede En una función con dominio y output acotado si se	0.5 Bien No se contestas la	Falta explicitar que el % de coverage no garantiza 0 calidad	nombraron 0 demasiados	4.85
16635442 16635469	1.5 ¡Perfecto!	No separa los tests minimos para cada tipo de Coverage. Estan bien los tests en global pero no me dice "test 1 2 3 4 para Statement	coverage, pero falló todo lo de branch coverage, (c.3.). También, ojo con la notación del grafo (c.1.). Cada bloque de oddigo es un nodo, y los controles de flujo definen las manso, no se debe analizar cada controles de flujo definen las comas, no se debe analizar cada controles por separado. 1 jiMuy bien!	En una función con dominio y output acotado si se 0.25 puede En una función con dominio y output acotado si se	0.5 Bien No se contestas la	Falta explicitar que el % de coverage no garantiza 0 calidad	nombraron 0 demasiados	4.85
16635442 16635469	1.5 ¡Perfecto!	No separa los tests minimos para cada tipo de Coverage. Estan bien los tests en global pero no me dice "test 1 2 3 4 para Statement	coverage, pero falló todo lo de branch coverage, (0.3). También, ojo con la notación del grafo (0.1). Cada bloque de código es un nodo, y los controles de flujo definen las ramas, no se debe analizar cada 0.5 confición por separado. 1 Muy bien! Bien! Pero hay das cosas (y son un tecnisismo, por lo que no hay descuento). I. El nodo de else-if no descuento. I. El nodo de else-if no	En una función con dominio y output acotado si se 0.25 puede En una función con dominio y output acotado si se	0.5 Bien No se contestas la	Falta explicitar que el % de coverage no garantiza 0 calidad	nombraron 0 demasiados	4.85
16635442 16635469	1.5 ¡Perfecto!	No separa los tests minimos para cada tipo de Coverage. Estan bien los tests en global pero no me dice "test 1 2 3 4 para Statement	coverage, pero falló todo lo de branch coverage, (0.3). También, ojo con la notación del grafo (0.1). Cada bloque de código es un nodo, y los controles de flujo definen las ramas, no se debe analizar cada 0.5 condición por separado. 1 Muy bien! Blen!. Pero hay dos cosas (y son un tecnisismo, por lo que no hay un tecnisismo de la que no hay un tecnision de la que no hay un tecnision de la	En una función con dominio y output acotado si se 0.25 puede En una función con dominio y output acotado si se	0.5 Bien No se contestas la	Falta explicitar que el % de coverage no garantiza 0 calidad	nombraron 0 demasiados	4.85
16635442 16635469	1.5 ¡Perfecto!	No separa los tests minimos para cada tipo de Coverage. Estan bien los tests en global pero no me dice "test 1 2 3 4 para Statement	coverage, pero falló todo lo de branch coverage, (0.3). También, ojo con la notación del grafo (0.1). Cada bloque de código es un nodo, y los controles de flujo definen las ramas, no se debe analizar cada 0.5 condición por separado. 1 Muy bien! Blen!. Pero hay dos cosas (y son un tecnisismo, por lo que no hay un tecnisismo de la que no hay un tecnision de la que no hay un tecnision de la	En una función con dominio y output acotado si se 0.25 puede En una función con dominio y output acotado si se	0.5 Bien No se contestas la	Falta explicitar que el % de coverage no garantiza 0 calidad	nombraron 0 demasiados	4.85
16635442 16635469	1.5 ¡Perfecto!	No separa los tests minimos para cada tipo de Coverage. Estan bien los tests en global pero no me dice "test 1 2 3 4 para Statement	coverage, pero falló todo lo de branch coverage, (0.3). También, ojo con la notación del grafo (0.1). Cada bloque de código es un nodo, y los controles de flujo definen las ramas, no se obee analizar cada 0.5 contidisón por separado. 1 Muy bien!	En una función con dominio y output acotado si se 0.25 puede En una función con dominio y output acotado si se	0.5 Bien No se contestas la	Falta explicitar que el % de coverage no garantiza 0 calidad	nombraron 0 demasiados	4.85
16635442 16635469	1.5 ¡Perfecto!	No separa los tests minimos para cada tipo de Coverage. Estan bien los tests en global pero no me dice "test 1 2 3 4 para Statement	coverage, pero falló todo lo de branch coverage, (0.3). También, ojo con la notación del grafo (0.1). Cada bloque de código es un nodo, y los controles de flujo definen las ramas, no se debe analizar cada 0.5 contidisón por separado. 1 Muy bien!	En una función con dominio y output acotado si se 0.25 puede En una función con dominio y output acotado si se	0.5 Bien No se contestas la	Falta explicitar que el % de coverage no garantiza 0 calidad	nombraron 0 demasiados	4.85
16635442 16635469	1.5 ¡Perfecto! 1.5 ¡Perfecto! Hay un test que dice fallar porque no	No separa los tests mínimos para cada tipo de Coverage. Estan bien los tests en global pero no me dice "test 1,2,3,4 para Statement coverage" rests 1,2,3,4,5 para 1 Branch" y assi	coverage, pero falló todo lo de branch coverage, (0.3). También, ojo con la notación del grafo (0.1). Cada bloque de código es un nodo, y los controles de flujo definen las manse, no se debe analizar cada contidente de controles de flujo definen las manse, no se debe analizar cada contidente por separado. 1 j.Muy bien! 2 j.Muy bien! 3 j.Muy bien! 4 j.Muy bien! 2 j.Muy bien! 2 j.Muy bien! 3 j.Muy bien! 4 j.Muy bien! 4 j.Muy bien! 5 j.Muy bien! 5 j.Muy bien! 5 j.Muy bien! 6 j.Muy bien!	En una función con dominio y output acotado si se 0.25 puede En una función con dominio y output acotado si se	0.5 Bien No se contestas la	Falta explicitar que el % de coverage no garantiza 0 calidad	nombraron 0 demasiados	4.85
16635442 16635469	1.5 ¡Perfecto! 1.5 ¡Perfecto! Hay un test que dice fallar porque no tiene diglio pero si lo tiene (el cuarto	No separa los tests mínimos para cada tipo de Coverage. Estan bien los tests en global pero no me dice "test 1,2.3 4,3 para Statement coverage" "tests 1,2.3 4,5 para ficinom" y así	coverage, pero falló todo lo de branch coverage, (c.3.). También, ojo con la notación del grafo (c.1.). Cada bloque de código es un nodo. Cada bloque de código es un nodo. Iramas, no se debe analizar cada 0.6 condición por separado. 1 jMuy bien! 1 jMuy bien! 1 jBien!. Pero hay dos cosas (y son un tecnisismo, por lo que no hay un tecnisismo, por lo que no hay completo es encesario, porque el control de flujo completo es un solo nodo (del fl debieran salla "Caminos). 2. Conditional coverage pide que cada condición bodecian sea truefalse al menos uma vez por a jeda, por lo combinatória y el porcentaje cambia	En una función con dominio y output acotado si se 0.25 puede En una función con dominio y output acotado si se	0.5 Bien No se contestas la	Falta explicitar que el % de coverage no garantiza 0 calidad	nombraron 0 demasiados	4.85
16835442 16835469 16835485	1.5 ¡Perfecto! 1.5 ¡Perfecto! Hay un test que dice fallar porque no tiene digito pero si lo liene del carto test). Falla proque no cumple el largo mínimo, pero eso y acetáen otro	No separa los tests mínimos para cada tipo de Coverage. Estan bien los tests en global pero no me dice "test 1,2,3,4 para Statement coverage" tests 1,2,3,4,5 para 1 Branch" — y ast 1 para de la franch " — y ast 1 par	coverage, pero falló todo lo de branch coverage, (c.3.). También, ojo con la notación del grafo (c.1.). Cada bloque de oddigo es un nodo, y mo controles de flujo definen las controles de controles	En una función con dominio y output acotado si se puede En una función con dominio y output acotado si se 0.25 podría	0.5 Bien No se contestas la 0.25 segunda pregunta	Falta explicitar que el % de coverage no garantiza calificad calific	nombraron 0 demasiados 0.5 Bien	4.85 6.00 1.00
16635442 16635469	1.5 ¡Perfecto! 1.5 ¡Perfecto! Hay un test que dice fallar porque no tiene diglio pero si lo tiene (el cuarto test). Eale porque no cumple el largo	No separa los tests mínimos para cada tipo de Coverage. Estan bien los tests en global pero no me dice "test 1,2,3,4 para Stamement coverage" flests 1,2,3,4,5 para 1 Branch" y asis	coverage, pero falló todo lo de branch coverage, (0.3). También, ojo con la notación del grafo (0.1). Cada bloque de código es un nodo, y los controles de flujo definen las armas, no se debe analizar cada controles de flujo definen las armas, no se debe analizar cada controles por separado. 1 [Muy bien! 1 [Muy bien! 1 [Bien! Pero hay das cosas (y son un tecnisismo, por lo que no hay descuento). 1. El nodo de else-if no es necesario, porque el control de flujo compileto es un solo nodo (del if debieran salir 3 caminos). 2. Conditional coverage pide que cada condición booleana sea true false al memos uma vez por al sola, por lo combinatoria y el porcentaje cambia (en este no hay descuentaje cambia (en este no hay descuenta porque	En una función con dominio y output acotado si se 0.25 puede En una función con dominio y output acotado si se	0.5 Bien No se contestas la	Falta explicitar que el % de coverage no garantiza 0 calidad	0 demassados 0.5 Bien 0.5 Bien	4.85
16635442 16635469 16635485	1.5 ¡Perfecto! 1.5 ¡Perfecto! Hay un test que dice fallar porque no tene diglio pero si lo tiene (el cuanto test). Polla proque no cumple el largo mino, pero eso ya está en otro 1.4 test.	No separa los tests mínimos para cada tipo de Coverage. Estan bien los tests en global pero no me dice "test 1,2,3,4 para Statement coverage" tests 1,2,3,4,5 para 1 Branch" — y ast 1 para de la franch " — y ast 1 par	coverage, pero falló todo lo de branch coverage, (c.3.). También, ojo con la notación del grafo (c.1.). Cada bloque de oddigo es un nodo, y mo controles de flujo definen las controles de controles	En una función con dominio y output acotado si se puede En una función con dominio y output acotado si se 0.25 podría 20.25 podría	0.5 Bien No se contestas la 0.25 segunda pregunta	Faita explicitar que el % de coverage no garantiza o calidad	nombraron 0 demassados 0.5 Bien 0.5 Bien	4.85 6.00 1.00
16635442 16635469 16635485	1.5 ¡Perfecto! 1.5 ¡Perfecto! Hay un test que dice fallar porque no tiene digito pero si lo liene del carto test). Falla proque no cumple el largo mínimo, pero eso y acetáen otro	No separa los tests mínimos para cada tipo de Coverage. Estan bien los tests en global pero no me dice "test 1,2,3,4 para Statement coverage" tests 1,2,3,4,5 para 1 franch" — y ast 1 franch" — y ast 1 franch —	coverage, pero falló todo lo de branch coverage, (0.3). También, ojo con la notación del grafo (0.1). Cada bloque de código es un nodo, y los controles de flujo definen las armas, no se debe analizar cada controles de flujo definen las armas, no se debe analizar cada controles pero esta de controles de flujo definen las armas, no se debe analizar cada controles pero esta de controles que no hay descuento). Le llondo de else-if no es necesario, porque el control de flujo completo es un solo nodo (del if debieran salir 3 caminos). 2. Conditional coverage pide que cada controles pero esta de controles de contro	En una función con dominio y output acotado si se puede En una función con dominio y output acotado si se 0.25 podría	0.5 Bien No se contestas la 0.25 segunda pregunta	Falta explicitar que el % de coverage no garantiza calificad calific	0 demassados 0.5 Bien 0.5 Bien	4.85 6.00 1.00
16635442 16635469 16635485	1.5 ¡Perfecto! 1.5 ¡Perfecto! Hay un test que dice fallar porque no tene diglio pero si lo tiene (el cuanto test). Polla proque no cumple el largo mino, pero eso ya está en otro 1.4 test.	No separa los tests mínimos para cada tipo de Coverage. Estan bien los tests en global pero no me dice "test 1,2,3,4 para Statement coverage" tests 1,2,3,4,5 para 1 franch" — y ast 1 franch" — y ast 1 franch —	coverage, pero falló todo lo de branch coverage, (0.3). También, ojo con la notación del grafo (0.1). Cada bloque de código es un nodo, y los controles de flujo definer las ramas, no se debe analizar cada o.5 controles de flujo definer las ramas, no se debe analizar cada o.5 controles por separado. 1 [Muy bien!	En una función con dominio y output acotado si se puede En una función con dominio y output acotado si se 0.25 podría 20.25 podría	0.5 Bien No se contestas la 0.25 segunda pregunta	Faita explicitar que el % de coverage no garantiza o calidad	nombraron 0 demassados 0.5 Bien 0.5 Bien	4.85 6.00 1.00
16635442 16635469 16635485	1.5 ¡Perfecto! 1.5 ¡Perfecto! Hay un test que dice fallar porque no tene diglio pero si lo tiene (el cuanto test). Polla proque no cumple el largo mino, pero eso ya está en otro 1.4 test.	No separa los tests mínimos para cada tipo de Coverage. Estan bien los tests en global pero no me dice "test 1,2,3,4 para Statement coverage" tests 1,2,3,4,5 para 1 franch" — y ast 1 franch" — y ast 1 franch —	coverage, pero falló todo lo de branch coverage, (0.3). También, ojo con la notación del grafo (0.1). Cada bloque de código es un nodo, y los controles de flujo definen las 0.6 condición por separado. 1 jMuy bien! Bient. Pero hay dos cosas (y son un tecnisismo, por lo que no hay descuento). Le li nodo de else-if no es necesario, portue de control de flujo completo es un solo nodo del portugue de control de flujo completo es un solo nodo del condición bodocana sea truefalse al menos una vez por si sola, por lo que no hay que hacer la combinatoria y el porcentaje cambia (en este no hay descuento). La combiento porque no es y que hacer la combinatoria y el porcentaje cambia (en este no hay descuento porque no es materia ofical y otras fuentes pueden explicitario diferente). 1 jMuy bien!	En una función con dominio y output acotado si se puede En una función con dominio y output acotado si se 0.25 podría 20.25 podría	0.5 Bien No se contestas la 0.25 segunda pregunta	Faita explicitar que el % de coverage no garantiza o calidad	nombraron 0 demassados 0.5 Bien 0.5 Bien	4.85 6.00 1.00
16635442 16635469 16635485	1.5 ¡Perfecto! 1.5 ¡Perfecto! Hay un test que dice fallar porque no tene diglio pero si lo tiene (el cuanto test). Polla proque no cumple el largo mino, pero eso ya está en otro 1.4 test.	No separa los tests mínimos para cada tipo de Coverage. Estan bien los tests en global pero no me dice "test 1,2,3,4 para Statement coverage" tests 1,2,3,4,5 para 1 franch" — y ast 1 franch" — y ast 1 franch —	coverage, pero falló todo lo de branch coverage, (0.3). También, ojo con la notación del grafo (0.1). Cada bloque de código es un nodo, y los controles de flujo definen las ramas, no se debe analizar cada 0.5 contidos por separado. 1 [Muy bien! 1 [Muy bien! 1 [Bient.] Pero hay des cosas (y son un tercisiamo, per lo que no hay descuento) 1. El nodo de else-if no es necesario, porque el control de flujo compileto es un solo nodo (del if debieran salla 3 caminos). 2. Conditional coverage pide que cada condición booleana sea truefalse al menos una vez por si sola, por lo que no hay que hacer ta combiantora y el porcentaje cambie combiento en y el porcentaje cambie combiento en y el porcentaje cambie no hay que hacer ta combiantora y el porcentaje cambie no es materia oficial y otras fuentes. 1 [Muy bien!	En una función con dominio y output acotado si se puede En una función con dominio y output acotado si se 0.25 podría 20.25 podría	0.5 Bien No se contestas la 0.25 segunda pregunta	Faita explicitar que el % de coverage no garantiza o calidad	nombraron 0 demassados 0.5 Bien 0.5 Bien	4.85 6.00 1.00
16635442 16635469 16635485	1.5 ¡Perfecto! 1.5 ¡Perfecto! Hay un test que dice fallar porque no tene diglio pero si lo tiene (el cuanto test). Polla proque no cumple el largo mino, pero eso ya está en otro 1.4 test.	No separa los tests mínimos para cada tipo de Coverage. Estan bien los tests en global pero no me dice "test 1,2,3,4 para Statement coverage" tests 1,2,3,4,5 para 1 franch" — y ast 1 franch" — y ast 1 franch —	coverage, pero falló todo lo de branch coverage, (e.3.). También, ojo con la notación del grafo (c.1.). Cada bloque de oddigo es un nodo. Cada bloque de oddigo es un nodo. In cada bloque de oddigo es un nodo. In cada bloque de cada cada (e. 1.). Cada bloque de cada (e. 1.). Cada bloque de cada (e. 1.). Cada (En una función con dominio y output acotado si se puede En una función con dominio y output acotado si se 0.25 podría 20.25 podría	0.5 Bien No se contestas la 0.25 segunda pregunta	Faita explicitar que el % de coverage no garantiza o calidad	nombraron 0 demassados 0.5 Bien 0.5 Bien	4.85 6.00 1.00
16635442 16635469 16635485	1.5 ¡Perfecto! 1.5 ¡Perfecto! Hay un test que dice fallar porque no tene diglio pero si lo tiene (el cuanto test). Polla proque no cumple el largo mino, pero eso ya está en otro 1.4 test.	No separa los tests mínimos para cada tipo de Coverage. Estan bien los tests en global pero no me dice "test 1,2,3,4 para Statement coverage" tests 1,2,3,4,5 para 1 franch" — y ast 1 franch" — y ast 1 franch —	coverage, pero falló todo lo de branch coverage, (e.3.). También, ojo con la notación del grafo (c.1.). Cada bloque de oddigo es un nodo. Cada bloque de oddigo es un nodo. In cada bloque de oddigo es un nodo. In cada bloque de cada cada (e. 1.). Cada bloque de cada (e. 1.). Cada bloque de cada (e. 1.). Cada (En una función con dominio y output acotado si se puede En una función con dominio y output acotado si se 0.25 podría 20.25 podría	0.5 Bien No se contestas la 0.25 segunda pregunta	Faita explicitar que el % de coverage no garantiza o calidad	nombraron 0 demassados 0.5 Bien 0.5 Bien	4.85 6.00 1.00
16635442 16635469 16635485	1.5 Perfectol 1.5 Perfectol Hay un test que dice fallar porque no tene digliu pero si o tiene (el cuarto test). Falla porque no cumple el largo mínimo, pero eso ya está en otro 1. let. 1.5 Perfectol	No separa los tests mínimos para cada tipo de Coverage. Estan bien los tests en global pero no me dice "test 1,2,3,4 para Statement coverage" tests 1,2,3,4,5 para 1 franch" — y ast 1 franch" — y ast 1 franch —	coverage, pero falló todo lo de branch coverage, (0.3). También, ojo con la notación del grafo (0.1). Cada bloque de código es un nodo, y los controles de flujo definen las armas, no se debe analizar cada controles de flujo definen las armas, no se debe analizar cada controles pero esta de controles de flujo definen las armas, no se debe analizar cada controles pero esta de flujo completo es un solo nodo (del if debieran salir 3 caminos). 2. Conditional coverage pide que cada condición booleana sea truefalse al controles pero esta de controles de flujo completo es un solo nodo (del fluj	En una función con dominio y output acotado si se puede En una función con dominio y output acotado si se 0.25 podría 20.25 podría	0.5 Bien No se contestas la 0.25 segunda pregunta	Faita explicitar que el % de coverage no garantiza o calidad	nombraron 0 demassados 0.5 Bien 0.5 Bien	4.85 6.00 1.00
16635442 16635469 16635485	1.5 ¡Perfectol 1.5 ¡Perfectol Hay un test que dice fallar porque no tiene diglio pero si lo tiene (el cuarto test). Falla porque no cumple el largo mínimo, pero eso ya está en otro 1.4 lest. 1.5 ¡Perfectol Hay un test que dice fallar porque no	No separa los tests mínimos para cada tipo de Coverage. Estan bien los tests en global pero no me dice "test 1,2,34 para Statement coverage" tests 1,2,34,5 para 1 Branch" — y asi 1 Branch" — y asi 1 Branch — y	coverage, pero falló todo lo de branch coverage, (c.3.). También, ojo con la notación del grafo (c.1.). Cada bloque de oddigo es un nodo, y los controles de flujo definen las controles controles de flujo definen las controles con	En una función con dominio y output acotado si se puede En una función con dominio y output acotado si se 0.25 podría 20.25 podría	0.5 Bien No se contestas la 0.25 segunda pregunta	Faita explicitar que el % de coverage no garantiza o calidad	nombraron 0 demassados 0.5 Bien 0.5 Bien	4.85 6.00 1.00
16635442 16635469 16635485	1.5 Perfectol 1.5 Perfectol 1.5 Perfectol 1.5 Perfectol Hay un test que dice fallar porque no tiene digno pero al o tiene de Losarto test). Esta porque no cumple el largo mínimo, pero eso ya está en otro 1.4 est. 1.5 Perfectol Hay un test que dice fallar porque no tiene diglio pero si o tiene el cuarto	No separa los tests mínimos para cada tipo de Coverage. Estan bien los tests en global pero no me dice "test 1,2.3 4 para Stamement coverage" tests 1,2.3 4,5 para francio" — y sai 1 francio" — y sai 1 francio" — y sai 1 francio	coverage, pero falló todo lo de branch coverage, (0.3). También, ojo con la notación del grafo (0.1). Cada bloque de código es un nodo, y los controles de flujo definen las descripcios de controles de flujo definen las descripcios de controles de flujo definen cada (0.6). Condición por separado. 1 j.Muy bien! 2 j.Muy bien de control de flujo compieto es un solo nodo (del lí debieran sala? 3 caminos). 2 condición booleana sea truefalse al menos una vez por si sola, por lo que no hay que hacer la combinatoria y el porcentaje cambia (en este no hay descuento). 1 j.Muy bien! 1 j.Muy bien! 1 j.Muy bien! 1 j.Muy bien! 2 j.Muy bien! 3 j.Muy bien! 3 j.Muy bien! 3 j.Muy bien! 3 j.Muy bien! 4 j.Muy bien! 4 j.Muy bien! 5 j.Conditional coverage pide que cada condición booleana sea truefalse al menos un avez por si sola, por lo que no hay que control de flujo compieto es un solo nodo (del lí debieran sala? 3 caminos). 2. Conditional coverage pide que cada condición booleana sea truefalse al menos una vez por si sola, por lo que no hay que hacer la que cambia condición booleana sea truefalse al menos una vez por si sola, por lo que no hay que hacer la que cambia condición booleana sea truefalse al menos una vez por si sola, por lo que no hay descuento porque in central por condición porque no po	En una función con dominio y output acotado si se puede En una función con dominio y output acotado si se 0.25 podría 20.25 podría	0.5 Bien No se contestas la 0.25 segunda pregunta	Faita explicitar que el % de coverage no garantiza o calidad	nombraron 0 demassados 0.5 Bien 0.5 Bien	4.85 6.00 1.00
16635442 16635469 16635485 16635582 16635787	1.5 Perfectol 1.5 Perfectol 1.5 Perfectol 1.5 Perfectol Hay un test que dice fallar porque no tiene digito pero si lo tiene (et canto test). Falla porque no cumple el largo mínimo, pero eso ya está en otro 1.4 lest. 1.5 Perfectol Hay un test que dice fallar porque no tiene digito pero si lo tiene (et cuanto tiene digito pero si lo tiene (et cuanto cominimo, pero eso ye está en otro	No separa los tests mínimos para cada tipo de Coverage. Estan bien los tests en global pero no me dice "test 1,2.3,4 para Statement coverage" tests 1,2.3,4,5 para francio" — y ast 1 francio	coverage, pero falló todo lo de branch coverage, (0.3). También, ojo con la notación del grafo (0.1). Cada bloque de código es un nodo, y los controles de flujo definen las descripcios de controles de flujo definen las descripcios de controles de flujo definen cada (0.6). Condición por separado. 1 j.Muy bien! 2 j.Muy bien de control de flujo compieto es un solo nodo (del lí debieran sala? 3 caminos). 2 condición booleana sea truefalse al menos una vez por si sola, por lo que no hay que hacer la combinatoria y el porcentaje cambia (en este no hay descuento). 1 j.Muy bien! 1 j.Muy bien! 1 j.Muy bien! 1 j.Muy bien! 2 j.Muy bien! 3 j.Muy bien! 3 j.Muy bien! 3 j.Muy bien! 3 j.Muy bien! 4 j.Muy bien! 4 j.Muy bien! 5 j.Conditional coverage pide que cada condición booleana sea truefalse al menos un avez por si sola, por lo que no hay que control de flujo compieto es un solo nodo (del lí debieran sala? 3 caminos). 2. Conditional coverage pide que cada condición booleana sea truefalse al menos una vez por si sola, por lo que no hay que hacer la que cambia condición booleana sea truefalse al menos una vez por si sola, por lo que no hay que hacer la que cambia condición booleana sea truefalse al menos una vez por si sola, por lo que no hay descuento porque in central por condición porque no po	En una función con dominio y output acotado si se puede En una función con dominio y output acotado si se 0.25 podría el producto el produ	0.5 Bien No se contestas la 0.25 segunda pregunta 0.5 Bien 0.5 Bien	Falta explicitar que el % de coverage no garantiza caldises 0.5 Bien 0.5 Bien 0.5 Bien	0 demassados 0.5 Bien 0.5 Bien 0.5 Bien Falla enfoque 0.23 regresión	6.00 1.00 6.65 6.75
16635442 16635469 16635485	Hay un test que dice fallar porque no tiene diglio pero si o tiene (el cuarto test). Falla porque no cumple el largo minimo, pero eso ya está en otro 1.5 Perfectol 1.5 Perfectol Hay un test que dice fallar porque no tiene diglio pero si o tiene (el cuarto test). Falla porque no cumple el largo minimo, pero eso ya está en otro 1.5 Perfectol	No separa los tests mínimos para cada tipo de Coverage. Estan bien los tests en global pero no me dice "test 1,2,34 para Statement coverage" tests 1,2,34,5 para 1 franch" — y asis 1,2,34,5 para 1 franch — y asis 1,2,	coverage, pero falló todo lo de branch coverage, (0.3). También, ojo con la notación del grafo (0.1). Cada bloque de código es un nodo, y los controles de flujo definen las substancia de composito de controles de flujo definen las substancia de controles de flujo definen las substancia de controles de flujo completo es un solo nodo (del fl debieran sall 3 caminos). 2. Controles de flujo completo es un solo nodo (del fl debieran salla 3 caminos). 2. Conditional coverage pide que cada condición booleana sea truefalse al menos una vez por si sola, por lo combinatoria y el porcentaje cambia (en este no hay descuente porque no es materia oficial y otras fuentes.)	En una función con dominio y output acotado si se puede En una función con dominio y output acotado si se 0.25 podría 20.25 podría	0.5 Bien No se contestas la 0.25 segunda pregunta	Faita explicitar que el % de coverage no garantiza o calidad	nombraron 0 demassados 0.5 Bien 0.5 Bien	4.85 6.00 1.00
16635442 16635469 16635485 16635582 16635787	1.5 Perfectol 1.5 Perfectol 1.5 Perfectol 1.5 Perfectol Hay un test que dice fallar porque no tiene digito pero si lo tiene (et canto test). Falla porque no cumple el largo mínimo, pero eso ya está en otro 1.4 lest. 1.5 Perfectol Hay un test que dice fallar porque no tiene digito pero si lo tiene (et cuanto tiene digito pero si lo tiene (et cuanto cominimo, pero eso ye está en otro	No separa los tests mínimos para cada tipo de Coverage. Estan bien los tests en global pero no me dice "test 1,2.3,4 para Statement coverage" tests 1,2.3,4,5 para francio" — y ast 1 francio	coverage, pero falló todo lo de branch coverage, (c.3.). También, ojo con la notación del grafo (c.1.). Cada bloque de oddigo es un nodo. Cada bloque de oddigo es un nodo. I amas, no se debe analizar cada (c. 1.). Cada bloque de composito de condición por separado. 1 jMuy bien! [Bien!. Pero hay dos cosas (y son un tecnisismo, por lo que no hay descuerio). I. El nodo de else-il no flue composito de condición por separado. La composito de condición bolocardo de condición bolocardo de condición bolocardo de condición bolocardo se a tender de condición bolocardo se a tender de control de la deberar las al Caminos, 2. 2. Conditional coverage pide que cada condición bolocardo se a tender de cardo condici	En una función con dominio y output acotado si se puede En una función con dominio y output acotado si se 0.25 podría el producto el produ	0.5 Bien No se contestas la 0.25 segunda pregunta 0.5 Bien 0.5 Bien	Falta explicitar que el % de coverage no garantiza caldises 0.5 Bien 0.5 Bien 0.5 Bien	0 demassados 0.5 Bien 0.5 Bien 0.5 Bien Falla enfoque 0.23 regresión	6.00 1.00 6.65 6.75
16635442 16635469 16635485 16635582 16635787	1.5 Perfectol 1.5 Perfectol 1.5 Perfectol 1.5 Perfectol Hay un test que dice fallar porque no tiene digito pero si lo tiene (et canto test). Falla porque no cumple el largo mínimo, pero eso ya está en otro 1.4 lest. 1.5 Perfectol Hay un test que dice fallar porque no tiene digito pero si lo tiene (et cuanto tiene digito pero si lo tiene (et cuanto cominimo, pero eso ye está en otro	No separa los tests mínimos para cada tipo de Coverage. Estan bien los tests en global pero no me dice "test 1,2.3,4 para Statement coverage" tests 1,2.3,4,5 para francio" — y ast 1 francio	coverage, pero falló todo lo de branch coverage, (c.3.). También, ojo con la notación del grafo (c.1.). Cada bloque de oddigo es un nodo, y los centroles de flujo definen las controles controles de flujo definen las controles con	En una función con dominio y output acotado si se puede En una función con dominio y output acotado si se 0.25 podría 20.25 Bien 20.5 Bien	0.5 Bien No se contestas la 0.25 segunda pregunta 0.5 Bien 0.5 Bien	Falta explicitar que el % de coverage no garantiza caldises 0.5 Bien 0.5 Bien 0.5 Bien	0 demassados 0.5 Bien 0.5 Bien 0.5 Bien Falla enfoque 0.23 regresión	6.00 1.00 6.65 6.75
16635442 16635469 16635485 16635582 16635787	1.5 Perfectol 1.5 Perfectol 1.5 Perfectol 1.5 Perfectol Hay un test que dice fallar porque no tiene digito pero si lo tiene (et canto test). Falla porque no cumple el largo mínimo, pero eso ya está en otro 1.4 lest. 1.5 Perfectol Hay un test que dice fallar porque no tiene digito pero si lo tiene (et cuanto tiene digito pero si lo tiene (et cuanto cominimo, pero eso ye está en otro	No separa los tests mínimos para cada tipo de Coverage. Estan bien los tests en global pero no me dice "test 1,2.3,4 para Statement coverage" tests 1,2.3,4,5 para franción" — y así franción" — y así franción — y así 1.5	coverage, pero falló todo lo de branch coverage, (c.3.). También, ojo con la notación del grafo (c.0.1). Cada bloque de código es un nodo, y los controles de flujo definen las manas, nos el debe analizar cada comisión por separado. 1 j.Muy bien! 2 j.Muy bien de conserva de control de flujo completo es un solo nodo (del jí debieran salir 3 caminos). 2. Conditional coverage pide que cada control de flujo completo es un solo nodo (del jí debieran salir 3 caminos). 2. Conditional coverage pide que cada control de flujo completo es un solo nodo (del jí debieran salir 3 caminos). 2. Conditional coverage pide que cada control de flujo completo proque no es materia oficial y otras fuentes lo que no hay que hacer la combinatoria y el porcentaje cambia (en este no hay descuento porque no es materia oficial y otras fuentes la pueden explicitario diferente). 1 j.Muy bien! 2 j. Conditional coverage pide que cada condición booleana sea truefalse al menos una vez por si sola, por lo que no hay que hacer la combination y el porcentaje cambia (en este no hay descuento porque no es materia folcal y toras fuentes de menos una vez por si sola, por lo que no hay que hacer la combination y el porcentaje cambia (en este no hay descuento porque no es materia folcal y toras fuentes de menos una vez por si sola, por lo que no hay que hacer la combination y el porcentaje cambia (en este no hay descuento porque no es materia folcal y toras fuentes de menos una vez por si sola, por lo que no hay que hacer la combination y el porcentaje cambia (en este no hay descuento porque no es materia folcal y toras fuentes de menos una vez por si sola, por lo que no hay que hacer la combination y el porcentaje cambia (en este no hay descuento porque no es materia folcal y toras fuentes de menos una vez por si sola, por lo que no hay que hacer la combination y el porcentaje cambia (en este no hay desc	En una función con dominio y output acotado si se 0.25 puede En una función con dominio y output acotado si se 0.25 podría si se 0.25 podr	0.5 Bien No se contestas la 0.25 segunda pregunta 0.5 Bien 0.5 Bien	Falta explicitar que el % de coverage no garantiza caldises 0.5 Bien 0.5 Bien 0.5 Bien	0 demassados 0.5 Bien 0.5 Bien 0.5 Bien Falla enfoque 0.23 regresión	6.00 1.00 6.65 6.75
16635442 16635469 16635485 16635582 16633582	1.5 ¡Perfectol 1.5 ¡Perfectol 1.5 ¡Perfectol 1.5 ¡Perfectol Hay un test que dice fallar porque no tene digito pero si o tiene (el cuarto test). Falla porque no cumple el largo mon. pero eso ya está en otro 1.4 test. 1.5 ¡Perfectol Hay un test que dice fallar porque no tene digito pero si o tiene (el cuarto test). Falla porque no cumple el largo mínimo, pero eso ya está en otro 1.4 test.	No separa los tests mínimos para cada tipo de Coverage. Estan bien los tests en global pero no me dice "test 1,2.3,4 para Statement coverage" tests 1,2.3,4,5 para franción" — y así franción" — y así franción — y así 1.5	coverage, pero falló todo lo de branch coverage, (c.3.). También, ojo con la notación del grafo (c.1.). Cada bloque de oddigo es un nodo, y los controles de flujo definen las controles de flujo de	En una función con dominio y output acotado si se e e e e e e e e e e e e e e e e e	0.5 Bien No se contestas la 0.25 segunda pregunta 0.5 Bien 0.5 Bien	Falta explicitar que el % de coverage no garantiza caldises 0.5 Bien 0.5 Bien 0.5 Bien	0 demassados 0.5 Bien 0.5 Bien 0.5 Bien Falla enfoque 0.23 regresión	6.00 1.00 6.65 6.75
16635442 16635469 16635485 16635582 16633582	1.5 Perfectol 1.5 Perfectol 1.5 Perfectol 1.5 Perfectol Hay un test que dice fallar porque no tiene digito pero si lo tiene (et canto test). Falla porque no cumple el largo mínimo, pero eso ya está en otro 1.4 lest. 1.5 Perfectol Hay un test que dice fallar porque no tiene digito pero si lo tiene (et cuanto tiene digito pero si lo tiene (et cuanto cominimo, pero eso ye está en otro	No separa los tests mínimos para cada tipo de Coverage. Estan bien los tests en global pero no me dice "test 1,2.3,4 para Statement coverage" tests 1,2.3,4,5 para franción" — y así franción" — y así franción — y así 1.5	coverage, pero falló todo lo de branch coverage, (c.3.). También, ojo con la notación del grafo (c.0.1). Cada bloque de código es un nodo, y los controles de flujo definen las manas, nos el debe analizar cada comisión por separado. 1 j.Muy bien! 2 j.Muy bien de conserva de control de flujo completo es un solo nodo (del jí debieran salir 3 caminos). 2. Conditional coverage pide que cada control de flujo completo es un solo nodo (del jí debieran salir 3 caminos). 2. Conditional coverage pide que cada control de flujo completo es un solo nodo (del jí debieran salir 3 caminos). 2. Conditional coverage pide que cada control de flujo completo proque no es materia oficial y otras fuentes lo que no hay que hacer la combinatoria y el porcentaje cambia (en este no hay descuento porque no es materia oficial y otras fuentes la pueden explicitario diferente). 1 j.Muy bien! 2 j. Conditional coverage pide que cada condición booleana sea truefalse al menos una vez por si sola, por lo que no hay que hacer la combination y el porcentaje cambia (en este no hay descuento porque no es materia folcal y toras fuentes de menos una vez por si sola, por lo que no hay que hacer la combination y el porcentaje cambia (en este no hay descuento porque no es materia folcal y toras fuentes de menos una vez por si sola, por lo que no hay que hacer la combination y el porcentaje cambia (en este no hay descuento porque no es materia folcal y toras fuentes de menos una vez por si sola, por lo que no hay que hacer la combination y el porcentaje cambia (en este no hay descuento porque no es materia folcal y toras fuentes de menos una vez por si sola, por lo que no hay que hacer la combination y el porcentaje cambia (en este no hay descuento porque no es materia folcal y toras fuentes de menos una vez por si sola, por lo que no hay que hacer la combination y el porcentaje cambia (en este no hay desc	En una función con dominio y output acotado si se 0.25 puede En una función con dominio y output acotado si se 0.25 podría si se 0.25 podr	0.5 Bien No se contestas la 0.25 segunda pregunta 0.5 Bien 0.5 Bien	Falta explicitar que el % de coverage no garantiza caldises 0.5 Bien 0.5 Bien 0.5 Bien	0 demassados 0.5 Bien 0.5 Bien 0.5 Bien Falla enfoque 0.23 regresión	6.00 1.00 6.65 6.75

				<u>_</u>				
			Ojo con el grafo y los nodos, tienen					
			dos fragmentos de código idénticos (término con return True) y					
			aparecen como nodos distintos en					
			su grafo (-0.1), NOTA; que entrege					
16636600	¡Perfecto! Deben indicar su pareja en 1.5 el PDF	15	solo uno de la pareja para la 0.9 próxima, pues habrán descuentos.	0.5 Bien	0.5 Bien	0.5 Bien	0.25 Falta Regresión	6.6
10030000	1.5 et PDF	I.S	0.9 proxima, pues nabran descuentos.	U.S Biell	U.S Biell	U.S Biell	0.25 Falla Regresion	0.00
		No separa los tests minimos para cada tipo de Coverage. No estan						
		bien los tests en global (Hay uno						
		extra). Deberia estar organizado en		En una función con				
		"test 1,2,3,4 para Statement coverage" "tests 1,2,3,4,5 para	¡Sobresaliente! Muy buen uso del	dominio y output acotado si se	El costo monetario no es, solo la		Enfoque de	
16636732	1.5 ¡Perfecto!	1 Branch" y asi	1 grafo con cada caso	0.25 puede	0.25 madurez	0.5 Bien	0.25 regresión	5.75
10000102	1.5 0.000.	i Didion y doi	i graio con cada caso	En una función con	0.E0 Induite	0.0 Didi	U.EU TOGICOION	0
				dominio v output	No se habla de la			
				acotado si se	madurez del			
16636910	1.5 ¡Perfecto!	1.5 Muy claro el grafo con colores!	1 ¡Sobresaliente! Muy buen grafo	0.25 puede	0.25 proyecto	0.5 Bien	0.5 Bien	6.50
	Si bien se aprecia el entregar							
	muchos test por casos. Hacer un test							
	que evalúan, por ejemplo, que retorne False porque no tiene dígitos,		Hay un error en el análisis de					
	da el mismo beneficio que crear 4		conditional coverage (el alumno 2					
	inputs diferentes. Faltó incluir alguna		del conjunto 1 si hace que tarea >=	En una función con	La segunda parte			
	justificación que diferencia lo que se	Falta 1 test en Branch Coverage.	4 evalue a false). En el caso de tu	dominio y output	es porque el			
16637402	busca testar con los 4 diferentes 1.2 inputs.	Hay 2 tests extra en Condition	pseudo-código, ambos tienen 90%	acotado si se	proyecto no es	0.5 01	0.05 Esta 4-	5.00
10037402	1.2 Inputs.	1 Coverage	0.8 de conditional coverage (-0.2)	0.25 puede	0.25 largo	0.5 Bien	0.25 Fata regresión	5.25
		No separa los tests minimos para cada tipo de Coverage. Estan bien						
		los tests en global pero no me dice				Un mayor % de		
		"test 1,2,3,4 para Statement				coverage no		
		coverage" "tests 1.2.3.4.5 para	.1		[garantiza mayor		
16637593	1.5 ¡Perfecto!	1 Branch" y asi	1 ¡Muy bien!	0.5 Bien	0.5 Bien	0 calidad	0.25 Falta regresión	5.75
		No separa los tests minimos para						
		cada tipo de Coverage. No estan bien los tests en global (Hay uno						
		extra). Deberia estar organizado en		En una función con				
		"tact 1 2 3 4 para Statement		dominio y output	El costo monetario			
		coverage" "tests 1.2.3.4.5 para	¡Sobresaliente! Muy buen uso del	acotado si se	no es, solo la		Enfoque de	
16637666	1.5 ¡Perfecto!	1 Branch" y asi	1 grafo con cada caso	0.25 puede	0.25 madurez	0.5 Bien	0.25 regresión	5.75
		El test T1 esta demas para	¡Buena respuesta! Pero ojo con el					
	Faltó el test donde se espera True	Statement Coverage. El nodo N1 no deberia estar en el grafo ya que la	caso 3 del segundo conjunto, dado que NP < 3.8, no se alcanza a					
	cuando se cumplen todos los	asignacion buyable= =false es la	evaluar AS. Esto no afecta el					
16637860	1.25 requisitos.	1.25 misma para todos.	1 resultado final.	0.5 Bien	0.5 Bien	0.5 Bien	0.5 Bien	6.50
	Dada la justificación entregada,							
	algunos test no eran necesarios, en							
	otras palabras, el beneficio de							
	algunos test entregados ya es suplido por otros. Para que la función							
	retorne True, se deben cumplir todos							
	los requisitos. Probar en cada uno							
	que se retorne True es similar a							
	hacer un test que retorne True (dado que es un AND). Faltó, tal vez, incluir			En una función con				
	que es un AND). Faltó, tal vez, incluir			dominio y output				
16637879	una justificación de que probarás 1.2 casos bordes para llegar al True.	1.5	1 iMuv bien!	acotado si se 0.25 puede	0.25 Nombra dinero	0.5 Bien	0.5 Bien	6.20
1003/8/9		1.5	i įmuy bieni	0.25 puede	U.25 Nombra dinero	U.5 Bien	U.5 Bien	6.20
	Si bien se aprecia el entregar muchos test por casos. Hacer un test							
	que evalúan, por ejemplo, que							
	retorne False porque no tiene dígitos,		Hay un error en el análisis de					
	da el mismo beneficio que crear 4		conditional coverage (el alumno 2		l			
	inputs diferentes. Faltó incluir alguna	E-th- 4 tt Bt O	del conjunto 1 si hace que tarea >=	En una función con	La segunda parte			
	justificación que diferencia lo que se busca testar con los 4 diferentes	Falta 1 test en Branch Coverage.	4 evalue a false). En el caso de tu	dominio y output acotado si se	es porque el proyecto no es			
16638360	1.2 inputs.	Hay 2 tests extra en Condition 1 Coverage	pseudo-código, ambos tienen 90% 0.8 de conditional coverage (-0.2)	0.25 puede	0.25 largo	0.5 Bien	0.25 Fata regresión	5.25
	,		Buen análisis y presentación de los					
			casos, pero se pedía justificación					
			en base al grafo que no aparece					
			(-0.2). Ojo que el análisis de					
			condition coverage no es 100%					
			todas las posibles combinaciones		Se habla de tiempo			
			de cada cláusula, sino que todas		ejecución y no de			
			sean al menos una vez true/false.		madurez de		No está bien	
			No hay descuento pues no se les		proyecto en la		enfocada la	
16639103	1.5 ¡Perfecto!	1.5	0.8 pasó como materia oficial.	0.5 Bien	0 segunda parte	0.5 Bien	0 respuesta	5.80
			En general bien, pero ojo con la notación de grafos (-0.1), si bien se					
			notación de grafos (-0.1), si bien se entiende las anotaciones de lineas,	En una función con	El número de			
	Faltó el test donde se espera True		en rigor se deben representar	dominio y output	líneas está			
	cuando se cumplen todos los		bloques de código como nodos y	acotado si se	relacionado, pero		Falta enfoque	
16639332	1.25 requisitos.	1.5	0.9 control de flujo como aristas.	0.25 puede	0.25 no es el factor	0.5 Bien	0.25 regresión	5.90
				En una función con				
				dominio y output				
16639537	1.E (Derfecte)	1.5	1 (Many blood	acotado si se 0.25 puede	0.5 Bien	0.5 Bien	0.25 Do ro14-	
16639537	1.5 ¡Perfecto!	1.5	1 ¡Muy bien!	0.25 puede	U.5 Bien	U.5 Bien	0.25 De regresión	6.50
	Muy interesante la forma que enfrentaste el problema. Lo que si, la							
	idea era testar cuando la función							
	retorna True o False según las							
	restricciones. En tu caso, dividiste el							
	problema en 2 y generaste casos de							
	prueba para cada parte, pero faltó la unión de ambos para notar que							
	algunos test validaban tanto la							1
	condición de contraseña como lo de							1
	menos de 4 dígitos consecutivos. De							1
	menos de 4 dígitos consecutivos. De todas formas se entrega todo el							
	puntaje porque se puede apreciar			F 4				
	que cumple el objetivo de diseñar pruebas suficientes y mínimas (dado	Condition Coverage v Branch		En una función con dominio y output				
	tu forma de responder) para el	coverage no tienen los tests		acotado si se				
16640373	1.5 problema dado.	1 minimos.	1 iMuv bien!	0.25 puede	0.5 Bien	0.5 Bien	0.25 Enfoque regresión	6.00
			· [1) =				quo rogradion	0.0

16640888	1.5 ¡Perfecto!	No separa los tests minimos para cada tipo de Coverage. Estan bien los tests en global pero no me dice "test 1,2,3,4 para Statement coverage" Tests 1,2,3,4,5 para		0.5 Bien	0.5 Bien	Un mayor % de coverage no garantiza mayor	0.05 Falances (4)	5.75
10040000	1.5 Penecio:	1 Branch* y asi	1 Muy bien!	En una función con		0 calidad	0.25 Falta regresión	5.75
16641299	1.5 Perfecto!	1.5 Muy claro el grafo con colores!	1 Sobresaliente! Muy buen grafo	dominio y output acotado si se 0.25 puede	No se habla de la madurez del 0.25 proyecto	0.5 Bien	0.5 Bien	6.50
17636906	1.5 prenedo:	1.5 May clare et grate con colores:	1 Journal and State Control of the C	0.23 paede	0.23 proyecto	U.S Biell	0.3 Dieli	1.00
17636973	1.5 ¡Perfecto!	Falta test donde no tiene suficiente dinero en Branch Coverage. No debería estar solo en Condition 1.25 Coverage	¡Muy bien en general! Pero ojo con la notación del grafo (-1), faltó el nodo final (return). Buena observación acerca de como varía el coverage dependiendo del 0.9 pseudo código ').	En una función con dominio y output acotado si se 0.25 puede	Falta madurez del 0.25 provecto	0.5 Bien	0.5 Bien	6.15
17636973	1.5 [Periedio:	1.25 Coverage	No es necesario un grafo específico	0.25 puede	0.25 proyecto	U.S Biell	U.S Biell	0.15
17637104	Faltó el test de que falte minuscula 1.25 pero si tener mayuscula y número.	1.5	para el condition coverage, el otro grafo que hiciste está blen. Buen análisis del branch coverage, pero el 1ª ojo" del conditional no es 0.7 correcto (-0.3)	En una función con dominio y output acotado si se 0.25 juede	0.5 Bien	EI % de coverage no garantiza mayor 0 calidad	0 Falta regresión	5.20
			Buen análisis, pero en el grafo faltó explicitar que de los nodos tipo "aprueba = true/false" se llega a uno final "return aprueba" (no hay			No habla de que		
17637740	1.5 ¡Perfecto!	Falta test donde no tiene suficiente dinero en Branch Coverage. No debería estar solo en Condition 1.25 Coverage	descuento porque es un tecnisismo). Hubiese estado 100% correcto si en lugar de asignar a 1 aprueba, se retornaba true / false.	En una función con dominio y output acotado si se 0.25 puede	Si bien la cantidad de funcionalidades esta correlacionado 0.25 no es la viariable	depende ya que un % de coverage mayor no garantiza 0 mejor codigo	Falta sistema e 0.25 integración	5.50
17637856	1.5 Perfecto!	15	1 iMuv bien!	En una función con dominio y output acotado si se 0.25 puede	Se habla de costos generales, la segunda relacionar con madurez de 0 provecto	0.5 Bien	Las más importantes son de sistemas e e 0.25 integración	6.00
1763/836	Dada la justificación entregada, algunos test no eran necesarios, en otras palabras, el beneficio de algunos test entregados ya es suplido por otros. Para que la función retorne True, se deben cumplir todos los requisitos. Probar en cada uno que se retorne True es similar a que se retorne True es similar a que se un AND). Faltó, tal vez, incluir una justificación de que probarás	15	1 Muy bien!	En una función con dominio y output acotado si se 0.25 puede	Falta explicictar madurez del proyecto en	Falta explicitar que el % de coverage no garantiza 0.25 calidad	Falta regresión sistema yo	
17638488	1.2 casos bordes para llegar al True.	1.5	Buen análisis, pero en el grafo faltó	U.25 puede	0.25 primera parte	0.25 calidad	0 integracion	5.45
17639182	1.5 Perfectol	Falta test donde no tiene sufficiente dinero en Branch Coverage. No debria estar solo en Condition 1.25 Coverage	explicitar que de los nodos tipo "aprueba = truefisale" se llega a uno final "return aprueba" (no hay descuento porque es un tecnisismo). Hubiese estado 100% correcto si en lugar de asignar a 1 aprueba, se retomaba true false.	En una función con dominio y output acotado si se 0.25 puede	Si bien la cantidad de funcionalidades esta correlacionado 0.25 no es la viariable	No habla de que depende ya que un % de coverage mayor no garantiza 0 mejor codigo	Falta sistema e 0.25 integración	5.50
		Fatan 2 tests en Branch coverage. Recordar que Statement es un subconjunto de Branch Coverage. No pueden haber mas tests en Statement Coverage que en	Buen análisis, pero faltó mostrar el	En una función con dominio y outputs acotado si se			Faltó mas precisión	
17640121	1.5 Perfecto!	1.25 Branch! Falta test donde no tiene suficiente dinero en Branch Coverage. No	grafo asociado (-0.2). jMuy bien en general! Pero ojo con la notación del grafo (-0.1), falló el nodo final (return). Buena observación acerca de como varía	0.25 puede En una función con dominio y output	0.5 Bien	0.5 Bien	0.25 en la respuesta	6.05
17640407	1.5 ¡Perfecto!	deberia estar solo en Condition 1.25 Coverage	el coverage dependiendo del 0.9 pseudo código :).	acotado si se 0.25 puede	Falta madurez del proyecto	0.5 Bien	0.5 Bien	6.15
17640598	1.5 ¡Perfecto!	Fatan 2 tests en Branch coverage. Recordar que Statement es un subconjunto de Branch Coverage. No pueden haber mas tests en Statement Coverage que en 1.25 Branch!	Buen análisis, pero faltó mostrar el 0.8 grafo asociado (-0.2).	En una función con dominio y outputs acotado si se 0.25 puede	0.5 Bien	0.5 Bien	Faltó mas precisión 0.25 en la respuesta	6.05
20405332	1.5 ¡Perfecto!	1.5	1 jMuy bien!	En una función con dominio y output acotado si se 0.25 puede	0.5 Bien	Hay más aspectos 0.25 a considerar	0.5 Bien	6.50
1562188J	1.5 Perfecto!	Hay un test extra en Condition 1.25 (Coverage, Deberian ser 6	Bien en general, pero si bien se entiende que NS representa a aprobado = True, en rigor no deben ser en mismo nodo (son bloques distintos de código, -0.1). Respecto al conditional coverage, ejo para el futuro que pide que cada statement booleano sea evaluada como true o false por si solo, no en base al 0.9 predicado compeleto.	En una función con dominio y output acotado si se 0.25 puede	0.5 Bien	0.5 Bien	Falto decir expliciamente 0.25i regresión	6.15
15021003	· · · · · · · · · · · · · · · · · · ·	Branch Coverage y Condition Coverage no leener el numero minimo de tests. Deberan ser 4 y 6	Us preocado compeno. Falto el grafo (> 0.2). La idea del branch coverage es que se analiza por las bituracciones posibles (P y T > 4.0 o T > 4.5, P > e.3 & y A > 0.9), por lo que escor casos que compeno de la propera el analisis (Falta aplican para el analisis (falta competamente en análisis de	En una función con dominio y output acotado si se	U.J. ORIII	U.S ORHI	v.co regresori	6.15
1563339J	1.5 ¡Perfecto!	1 respectivamente	0.5 branch coverage, -0.3)	0.25 puede	0.5 Bien	0.5 Bien	0.5 Bien	5.75
1563373J	1.5 ¡Perfecto!	No separa los tests minimos para cada tipo de Coverage. Estan bien los tests en global pero no me dice "test 1,2,3,4 para Statement coverage" "tests 1,2,3,4,5 para 1 Branch" y asi	1 ¡Muy bien!	En una función con dominio y output acotado si se 0.25 puede	Primera parte hablas de costo, 0.25 demasiado general	Un mayor % de coverage no garantiza mayor 0 calidad	No nombra 0.25 regresión	5.25

1563390J	Dada la justificación entregada, algunos test no eran necesarios, en otras palabras, el beneficio de algunos test entregados y anesón es es supido por otros. Para que la función retorne True, se deben cumplir tofos los requisitos. Probar en cada uno que se retorne True es similar a hacer un test que retorne True (dado que es un AND). Faltò, tal vez, incluir una justificación de que probarás 12 casos bordes para llegar al True.	Hay un test extra en Condition Coverage. Deberían ser 6 test 1.25 para esa coberfura.	Bien en general, pero falló el grafo (-0.2). Qio que condition coverage pide que cada condición booleana sea evaluada true y false, por lo que según tus análisis ambos tienen 100% condition coverage. No descuento porque e su no concepto que no abordamos en classes y tu 0.8 interpretación puede estar correcta.	En una función con dominio y output acctado si se 0.25 puede	0.5 Bien	0.5 Bien	El de usuario ya deberia haiber sido 0.25 testeado de antes	2 3.75
1563759J	Failtó el test donde se espera True cuando se cumplen todos los 1.25 requisitos.	El test 1 no deberia estar en Statement Coverage. Hay 2 te que son extra en Condition Coverage, deberia ser el minin 1 tests)	false, no en conjunto. No se	En una función con dominio y output acotado si se 0.25 puede	0.5 Bien	0.5 Bien	Enfoque de 0.25 regresión	5.75
1563941J	1.5 ¡Perfecto!	Hay un test extra en Condition Coverage. Deberian ser 6 test 1.25 para esa cobertura.	Blent Pero ojo con los conceptos para el branch coverage. La rama es de un nodo a otro, lo que mencionaron en su respuesta son 1 caminos.	0.5 Bien	0.5 Bien	0.5 Bien	0.25 Falta regresión	6.50
1663523J	1.5 ¡Perfecto!	Falta un test en Condition Coverage: Tiene dinero suficie no es recomendado, es el volu 1.25 1 y no tiene más de 7 volúmen Revisar las definiciones de cac	presentación de los casos, pero se s. 0.8 pedía justificación con grafo (-0.2).	0.5 Bien	0.5 Bien	0.5 Bien	Falta integración y sistemas	6.30
1663649J	1.5 ¡Perfectol	Revisar ias deminiciones de cati tipo de coverage. Hay un tests en condition coverage, deberá pueden haber mas tests minim en statement coverage que en branch coverage. Uno es "subconjunto" de otro. (Statemo 0.75 <= Branch)	otra ser Bien en generall Pero ojo con la s notación del grafo (-0.1), las condiciones se evalúan como un único nodo a pesar de tener	0.5 Bien	0.5 Blen	0.5 Bien	El enfoque más importante es de 0.25 regresión	5.90
1664171J	Faltó el test donde se espera True cuando se cumplen todos los 1.25 requisitos.	El test T1 esta demas para Statement Coverage. El nodo debería estar en el grafo ya qu asignacion buyable= =false es 1.25 misma para todos.	la que NP < 3.8, no se alcanza a	0.5 Bien	0.5 Bien	0.5 Bien	0.5 Bien	6.50
2010233J	Hay 2 test que revisan que retorne True si se cumplen todos los casos y 2 test que retornan False por tener 4 números consecutivos. Esos 4 test se pueden reducir a 2 test (uno con true y otro con false por los números 1.3 consecutivos).	Ese conjunto de datos no cum con todas los Coverago. Debe ser 3 para Statement Coverage para Branch Coverage y 6 pan 0.5 Condition Coverage	coverage es incorrecto (-0.3), se debe evaluar cuántos casos hay	En una función con dominio y output acotado si se 0.25 puede	Se habla de costo general, el costo es 0 la consecuencia	0.5 Bien	Fala enfoque regresión y sistema 0 y/o integración	4.05