N° Alumno	Diseño de pruebas	Comentario Diseño de pruebas	Coverage parte A	Comentario Coverage parte A	Coverage parte B	Comentario Coverage parte B	Pregunta Conceptual 1	Comentario P1	Pregunta Conceptual 2	Comentario P2	Pregunta Conceptual 3	Comentario P3	Pregunta Conceptual 4	Comentario P4	Descuento Atraso	Nota
12638781	1.5	¡Perfecto!	1.5		1	¡Muy bien!	0.25	En una función con dominio y output acotado si se puede	0	Se habla de costos generales, la segunda relacionar con madurez de proyecto	0.5	Bien	0.25	Las más importantes son de sistemas e integración		6.00
13620916	1.2	Hay test que evalúan lo mismo como el segundo y el último que fallan por no tener el largo mínimo. Uno de ellos no aporta más a detectar problemas.	1.25	Falta 1 test en Branch Coverage. (El caso 1 de Condition Coverage debería ser tambien de Branch Coverage)	0.3	Faltó el grafo (-0.2). Alumno 4 del conjunto 2 si entra al else if (falla parcial en análisis de branch coverage, -0.2). El conditional coverage no evalúa cada condición completa del if, sino que cada statement booleano por si solo (-0.3).	0.25	En una función con dominio y output acotado si se puede	0.25	No buena justifiación	0	Falto decir depende ya que un % de coverage no garantiza mejor calidad	0.25	Faltó enfoque regresión		4.50
13634372																1.00
13638386	1.25	Faltó el test donde se espera True cuando se cumplen todos los requisitos.	1.5		0.9	En general bien, pero ojo con la notación de grafos (-0.1), si bien se entiende las anotaciones de lineas, en rigor se deben representar bloques de código como nodos y control de flujo como aristas.	0.25	En una función con dominio y output acotado si se puede	0.25	El número de líneas está relacionado, pero no es el factor	0.5	Bien	0.25	Falta enfoque regresión		5.90
14619083	1.5	¡Perfecto!	1.25	Falta un test en condition coverage: Tiene dinero suficiente, no es recomendado, es el volumen 1 y tiene más de 7 volúmenes totales.	0.9	Bien en general, pero ojo con la notación del grafo (-0.1). Cada bloque de código es un nodo y cada condición es una bifuración. En tu caso, solo habrían 3 nodos y 3 ramas. El análisis de branch coverage está bien hecho bajo la lógica de tu grafo.	0.25	En una función con dominio y output acotado si se puede	0.5	Bien	0.5	Bien	0.25	Nombras demasiados en la segunda parte		6.15
14621150	1.5	¡Perfecto!	1.25		1	¡Bien! Pero ojo con los conceptos para el branch coverage. La rama es de un nodo a otro, lo que mencionaron en su respuesta son caminos.	0.5	Bien	0.5	Bien	0.5	Bien	0.25	Falta regresión		6.50
14632527	1.2	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 "si no se cae", pero esos test arrojan False porque no cumplen todos los requisitos para ser una contraseña segura. Además, para cada requisito agregaron otro test que efectivamente retorna True si se cumplen todos los requisitos, pero eso hace que testeen 5 veces el requisito de: retornar True si pasa todas las condiciones.	1	Statement Coverage no es minimo (Deberian ser 3 tests). Falta un test en Condition Coverage (revisar las condicions con OR)	1	¡Muy bien! Lo único es que para la otra consideren que los ands pueden no evaluarse si la claúsula anterior es False. No está mal, pero es un supuesto que deberían mencionar y que en este caso se infiere de su respuesta. Consideren que sin ese supuesto, el primer conjunto tiene menos condition coverage.	0.25	En una función con dominio y output acotado si se puede	0.5	Bien	0.5	Bien	0.5	Bien		5.95
14633426																1.00
14633450	1.5	Había un test que se podría haber omitido, que es el de más de 8 caracteres que retorna True porque el primer test ya valida que se cumpla el mínimo (8 o más), pero somo es solo un test, no se aplica descuento.	1.5		0.8	Buen análisis, pero faltó el grafo (-0.2)	0.25	En una función con dominio y output acotado si se puede	0.5	Bien	0.5	Bien	0.5	Bien	2	4.55
14636530	1.25	Faltó el test donde se espera True cuando se cumplen todos los requisitos.	1	El test 1 no deberia estar en Statement Coverage. Hay 2 tests que son extra en Condition Coverage, deberia ser el minimo (6 tests)	1	¡Muy bien! Eso si, el conditional coverage pide que cada statement booleano por si solo evalue a true y false, no en conjunto. No se descuenta porque la fuente puede ser válida igual.	0.25	En una función con dominio y output acotado si se puede	0.5	Bien	0.5	Bien	0.25	Enfoque de regresión		5.75
15204030	1.5	¡Perfecto!	0	No entrega pregunta	0	No entrega pregunta	0.25	En una función con dominio y output acotado si se puede	0.5	Bien	0.5	Bien	0.25	Falta enfoque regresión		4.00
15621146	1.5	¡Perfecto!	1	No separa Condition y Branch coverage. Se buscan los tests minimos para cada uno	1	¡Muy bien!	0.25	En una funció con dominio y ouput acotado si se puede	0.5	Bien	0.5	Bien	0	Mal enfoque y luego nivel tradicional		5.75
15621219	1.2	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 una justificación de que probarás casos bordes para llegar al True.	1	Branch Coverage y Condition Coverage no tienen el numero minimo de tests. Deberian ser 4 y 6 respectivamente	1	¡Muy bien!	0.25	En una función con dominio y output acotado si se puede	0.5	Bien	0	Se debe decir que depende, ya que el % no garantiza mejor codigo	0.5	Bien		5.45
15621278	1.25	Falta el test donde falte letra minúscula.	1	El numero de tests en Branch y Condition Coverage no son minimos.	1	Bien	0.25	En una función con dominio y output acotado si se puede	0.5	Bien	0.5	Bien	0.25	Falto decir explicitamente regresión		5.75
15621596	1.5	¡Perfecto!	1	No separa Condition y Branch coverage. Se buscan los tests minimos para cada uno.	1	¡Muy bien!	0.25	En una función con dominio y ouput acotado si se puede	0.5	Bien	0.5	Bien	0	Mal enfoque y luego nivel tradicional		5.75
15622355	1.5	¡Perfecto!	1	Falta 1 test en Branch Coverage. Hay 3 tests extra en Condition Coverage	0.5	Faltó el 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 >= 0.9), por lo que esos casos que mencionan que no se cubren no aplican para el análisis (falla completamente en análisis de branch coverage, -0.3)	0.25	En una función con dominio y output acotado si se puede	0.5	Bien	0.5	Bien	0.5	Bien		5.75

15623688	1.2	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 una justificación de que probarás casos bordes para llegar al True.	1	Falta 1 test en Branch Coverage. Hay 1 test extra en Condition Coverage	0.8	Buen análisis de branch coverage, pero al de condition le faltan números exactos de casos posibles y de casos logrados (-0.2)	0	No contesta	0	No contesta	0	No contesta	0	No contesta		4.00
15624005	1.25	Falta el test donde falte letra minúscula.	1	El numero de tests en Branch y Condition Coverage no son minimos.	1	Bien	0.25	En una función con dominio y output acotado si se puede	0.5	Bien	0.5	Bien	0.25	Falto decir explicitamente regresión		5.75
15633144	1.5	¡Perfecto!	1.25	Hay un test extra en Condition Coverage. Deberian ser 6	0.9	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 booleano sea evaluado como true o false por si solo, no en base al predicado completo.	0.25	En una función con dominio y output acotado si se puede	0.5	Bien	0.5	Bien	0.25	Falto decir explicitamente regresión		6.15
15633241	1.5	¡Perfecto!	1.25	Faltan 2 tests en Condition Coverage	0.8	Buen análisis, pero faltó el grafo (-0.2).	0.25	En una función con dominio y output acotado si se puede	0	Determina la madurez del proyecto	0.5	Bien	0.25	Fata de sistema o integracion		5.55
15633268	1.5	¡Perfecto!	1.25	Hay un test extra en Condition Coverage. Deberian ser 6 tests para esa cobertura.	1	¡Sobresaliente! Lo único es que para la otra consideren que los 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 tablas bien presentadas :). Consideren que sin ese supuesto, el primer conjunto tiene menos condition coverage.	0.25	En una función con dominio y output acotado si se puede	0.5	Bien	0.25	Un % de coverage no garantiza un mejor calidad de código	0	Falta nombrar regresión y hay algunos más importantes que otros		5.75
15634108	1.5	¡Perfecto!	1.25	Falta 1 test en Branch Coverage. (Test ID=4)	1	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.	0.25	En una función con dominio y output acotado si se puede	0.5	Bien	0.5	Bien	0.5	Bien		6.50
15634809	1.2	Solo era necesario 1 test por cada criterio que diga False y 1 test que diga True si pasa todos los test. Hay test redundantes cuya justificación es la misma.	1	Faltaron tests de las condiciones con "OR" en condition coverage.	0.7	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 de las condiciones C, D y F que eran la rama que seguía, entonces omitieron ciertos casos y sus análisis de condition coverage no están 100% correctos. (-0.3)	0.5	Bien	0.5	Bien	0.5	Bien	0.25	Falta regresión		5.65
15634884	1.2	Solo era necesario 1 test por cada criterio que diga False y 1 test que diga True si pasa todos los test. Hay test redundantes cuya justificación es la misma.	1	Faltaron tests de las condiciones con "OR" en condition coverage.	0.7	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 de las condiciones C, D y F que eran la rama que seguía, entonces omitieron ciertos casos y sus análisis de condition coverage no están 100% correctos. (-0.3)	0.5	Bien	0.5	Bien	0.5	Bien	0.25	Falta regresión		5.65
15635058	1.5	¡Perfecto!	1.25	Hay un test extra en Condition Coverage. Deberian ser 6 tests para esa cobertura.	1	¡Sobresaliente! Lo único es que para la otra consideren que los 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 tablas bien presentadas :). Consideren que sin ese supuesto, el primer conjunto tiene menos condition coverage.	0.25	En una función con dominio y output acotado si se puede	0.5	Bien	0.25	Un % de coverage no garantiza un mejor calidad de código	0	Falta nombrar regresión y hay algunos más importantes que otros		5.75
15635198	1.4	Agregaron un test de que string vacío era True, eso no cumple con las condiciones dadas por lo que es False. Todo lo demás está bien.	1.25	Falta 1 test de Condition Coverage.	0.9	Buen análisis 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, -0.1). Esto hizo que su branch coverage fuera igual a su conditional coverage.	0.25	En una función con dominio y output acotado si se puede	0.25	Si se tiene bajo presupueto, pero el proyecto es largo de todas maneras es más "barato"	0	La respuesta es que un % de coverage no garantiza calidad	0.25	No nombra regresión		5.30
15635465	1.2	Habían algunos test que evaluaban los mismo (como el de "seña" y "contra") o verificar con 1, 2, 3 y 4 digitos. Dada la explicación, era suficiente con 4 o más porque habían otros test que ya poseían menos de 4 dígitos. Esto hace que se tengan test redundantes.	1.25	Falta un test en condition coverage: Tiene dinero suficiente , no es un libro recomendado y si es el último volumen	0.8	Bien en general, pero ojo que condition coverage significa que cada condición booleana es al menos evaluada una vez como true y false, no es necesario que se considere un conjunto de estas. Faltó el grafo (-0.2, a pesar de que fuera simple, es necesario ver si pueden plasmar la representación)	0.5	Bien	0	No se habla de la madurez del proyecto	0	Un mayor % de coverage no garantiza mayor calidad	0.25	No se nombra regresión		5.00
15635635																1.00
15636143	1.5	¡Perfecto!	1	No separa los tests minimos para cada tipo de Coverage. No estan bien los tests en global (Hay uno extra). Deberia estar organizado en "test 1,2,3,4 para Statement coverage" "tests 1,2,3,4,5 para Branch" .. y asi	0.8	Análisis correcto, pero no hay pseudo-código (-0.2). Se infiere que es un es un if/elseif con returns directos.	0.25	En una función con dominio y output acotado si se puede	0.5	Bien	0	Un porcentaje mayor no garantiza una mayor calidad	0.25	De regresión		5.30

15636585	1	Una contraseña segura es aquella que cumple todos los requisitos, por lo tanto tienes test que esperan True cuando será False 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 "QWERTYUI" dices que es False porque no posee minúscula y si lo posee. Finalmente, se presentan muchos test que se invalidan por más de una restricción que no se justifica. Se entrega puntaje no completo porque en caso de fallar tus test test, no se sabría exactamente el origen de la falla.	1.25	Falta un test en condition coverage: Tiene dinero suficiente, no es recomendado, es el volumen 1 y no tiene más de 7	1	Bien en general. Pero ojo que conditional coverage significa que cada condición evalua true y false al menos una vez por si sola. Dado que no es materia que hayamos pasado, corregí en base a su suposición.	0	Puede pasar un test por suerte	0	No se habla de la madurez del proyecto	0.5	Bien	0.25	Enfoque de regresión		5.00
15637484	1.5	¡Perfecto!	1	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 Branch" .. y asi	1	¡Muy bien!	0.25	En una función con dominio y output acotado si se puede	0.25	Primera parte hablas de costo, demasiado general	0	Un mayor % de coverage no garantiza mayor calidad	0.25	No nombra regresión		5.25
15637638	1.2	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 una justificación de que probarás casos bordes para llegar al True.	1	No separa Condition y Branch coverage. Se buscan los tests minimos para cada uno	1	¡Muy bien!	0.25	En una función con dominio y output acotado si se puede	0.25	Se habla de costo general	0	Un % de coverage no garantiza un mejor código	0.25	Enfoque de regresión		4.95
15638634	1.4	Agregaron un test de que string vacío era True, eso no cumple con las condiciones dadas por lo que es False. Todo lo demás está bien.	1.25	Falta 1 test de Condition Coverage.	0.9	Buen análisis 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, -0.1). Esto hizo que su branch coverage fuera igual a su conditional coverage.	0.25	En una función con dominio y output acotado si se puede	0.25	Si se tiene bajo presupueto, pero el proyecto es largo de todas maneras es más "barato"	0	La respuesta es que un % de coverage no garantiza calidad	0.25	No nombra regresión		5.30
15638936	1.5	¡Perfecto!	0.5	Habia que separa los tests para cada tipo de coverage y obtener los tests minimos para cada uno	0.8	Buen análisis de branch coverage, pero al de condition le faltan números exactos de casos posibles y de casos logrados (-0.2)	0	A menos que sea un función con diminio y output acotados no se puede	0.5	Bien	0.5	Bien	0.25	Falto enfoque regresión	2	3.05
15638944	1	Una contraseña segura es aquella que cumple todos los requisitos, por lo tanto tienes test que esperan True cuando será False 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 "QWERTYUI" dices que es False porque no posee minúscula y si lo posee. Finalmente, se presentan muchos test que se invalidan por más de una restricción que no se justifica. Se entrega puntaje no completo porque en caso de fallar tus test test, no se sabría exactamente el origen de la falla.	1.25	Falta un test en condition coverage: Tiene dinero suficiente, no es recomendado, es el volumen 1 y no tiene más de 7	1	Bien en general. Pero ojo que conditional coverage significa que cada condición evalua true y false al menos una vez por si sola. Dado que no es materia que hayamos pasado, corregí en base a su suposición.	0	Puede pasar un test por suerte	0	No se habla de la madurez del proyecto	0.5	Bien	0.25	Enfoque de regresión		5.00
15638952	1.2	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 "si no se cae", pero esos test arrojan False porque no cumplen todos los requisitos para ser una contraseña segura. Además, para cada requisito agregaron otro test que efectivamente retorna True si se cumplen todos los requisitos, pero eso hace que testeen 5 veces el requisito de: retornar True si pasa todas las condiciones.	1	Statement Coverage no es minimo (Deberian ser 3 tests). Falta un test en Condition Coverage (revisar las condicions con OR)	1	¡Muy bien! Lo único es que para la otra consideren que los ands pueden no evaluarse si la claúsula anterior es False. No está mal, pero es un supuesto que deberían mencionar y que en este caso se infiere de su respuesta. Consideren que sin ese supuesto, el primer conjunto tiene menos condition coverage.	0.25	En una función con dominio y output acotado si se puede	0.5	Bien	0.5	Bien	0.5	Bien		5.95
15639053	1.5	¡Perfecto!	0.5	No separa los tests en cada tipo y no son minimos de Condition Coverage	0.9	Buen análisis 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, -0.1).	0.5	Bien	0	No se habla de la madurez del proyecto	0.5	Bien	0	Enfoque de regresión y test de sistemas e integración		4.90
15639495	1.5	¡Perfecto!	1	Falta 1 test en Branch Coverage y los tests en Condition Coverage no son los minimos (deberian ser 6)	0.8	Buen análisis en general, pero faltó rigor en los casos del conditional coverage (-0.2)	0.25	En una función con dominio y output acotado si se puede	0.25	No se contesta primera pregunta	0	Un mejor coverage no determina una mejor calidad	0	No contesta	2	2.80
15639746	1.5	¡Perfecto!	1.25	Falta 1 test en Branch Coverage. (Test ID=4)	1	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.	0.25	En una función con dominio y output acotado si se puede	0.5	Bien	0.5	Bien	0.5	Bien		6.50
16203615	1.2	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 una justificación de que probarás casos bordes para llegar al True.	1.25	Condition Coverage no tiene los tests minimos. (Hay 2 tests extras)	0.8	Buen análisis, pero faltó el grafo (-0.2).	0.5	Bien	0	No se habla de la madurez del proyecto	0	depende, % de coverage no garantiza calidad	0	Falta enfoque y no se especifica niveles tradicionales		4.75

16206851	1.5	¡Perfecto!	1.5		1	¡Muy bien!	0.25	En una función con dominio y output acotado si se puede	0.5	Bien	0.5	Bien	0.25	De regresión		6.50
16207041	1.5	¡Perfecto!	1.5		0.8	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 de cada cláusula, sino que todas sean al menos una vez true/false. No hay descuento pues no se les pasó como materia oficial.	0.5	Bien	0	Se habla de tiempo ejecución y no de madurez de proyecto en la segunda parte	0.5	Bien	0	No está bien enfocada la respuesta		5.80
16635019	1.5	¡Perfecto!	1	Statement Coverage no es mínimo (Deberían ser 3 tests). Falta un test en Condition Coverage (revisar las condicions con OR)	0.9	En general buen análisis, pero te equivocaste en el alumno 5 del conjunto 2, que va por el camino p3 y no p4. Esto hace que el branch coverage sea 100% y no 75% (-0.1).	0.5	Bien	0	No se habla de la madurez del proyecto	0.5	Bien	0.5	Bien		5.90
16635191	1.5	¡Perfecto!	1	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 Branch" .. y asi	1	¡Muy bien!	0.25	En una función con dominio y output acotado si se podría	0.25	No se contestas la segunda pregunta	0.5	Bien	0.5	Bien		6.00
16635221	1.5	¡Perfecto!	1.5		1	¡Muy bien!	0.5	Bien	0.5	Bien	0.5	Bien	0.25	Falta enfoque regresión		6.75
16635264	1.5	¡Perfecto!	0.75	Revisar las definiciones de cada tipo de coverage. Hay un tests extra en condition coverage, deberia ser el minimo numero de pruebas. No pueden haber mas tests minimos en statement coverage que en branch coverage. Uno es "subconjunto" de otro.(Statement <= Branch)	0.9	¡Bien en general! Pero ojo con la notación del grafo (-0.1), las condiciones se evalúan como un único nodo a pesar de tener múltiples statements booleanos, que simplifica bastante el análisis.	0.5	Bien	0.5	Bien	0.5	Bien	0.25	El enfoque más importante es de regresión		5.90
16635345	1.2	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 una justificación de que probarás casos bordes para llegar al True.	1	No separa Condition y Branch coverage. Se buscan los tests minimos para cada uno	1	¡Muy bien!	0.25	En una función con dominio y output acotado si se puede	0.25	Se habla de costo general	0	Un % de coverage no garantiza un mejor código	0.25	Enfoque de regresión		4.95
16635361																1.00
16635442	1.5	¡Perfecto!	1	Falta un test en Branch Coverage y en Condition Coverage	0.6	Bien el análisis de condition 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, y los controles de flujo definen las ramas, no se debe analizar cada condición por separado.	0.25	En una función con dominio y output acotado si se puede	0.5	Bien	0	Falta explicitar que el % de coverage no garantiza calidad	0	Faltó regresión y se nombraron demasiados		4.85
16635469	1.5	¡Perfecto!	1	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 Branch" .. y asi	1	¡Muy bien!	0.25	En una función con dominio y output acotado si se podría	0.25	No se contestas la segunda pregunta	0.5	Bien	0.5	Bien		6.00
16635485																1.00
16635582	1.4	Hay un test que dice fallar porque no tiene dígito pero si lo tiene (el cuarto test). Falla porque no cumple el largo mínimo, pero eso ya está en otro test.	1.25	Falta test donde no tiene suficiente dinero en Branch Coverage. No deberia estar solo en Condition Coverage	1	¡Bien!. Pero hay dos 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 completo es un solo nodo (del if debieran salir 3 caminos). 2. Conditional coverage pide que cada condición booleana sea true/false 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 porque no es materia oficial y otras fuentes pueden explicitarlo diferente).	0.5	Bien	0.5	Bien	0.5	Bien	0.5	Bien		6.65
16635787	1.5	¡Perfecto!	1.5		1	¡Muy bien!	0.5	Bien	0.5	Bien	0.5	Bien	0.25	Falta enfoque regresión		6.75
16636228	1.4	Hay un test que dice fallar porque no tiene dígito pero si lo tiene (el cuarto test). Falla porque no cumple el largo mínimo, pero eso ya está en otro test.	1.25	Falta test donde no tiene suficiente dinero en Branch Coverage. No deberia estar solo en Condition Coverage	1	¡Bien!. Pero hay dos 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 completo es un solo nodo (del if debieran salir 3 caminos). 2. Conditional coverage pide que cada condición booleana sea true/false 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 porque no es materia oficial y otras fuentes pueden explicitarlo diferente).	0.5	Bien	0.5	Bien	0.5	Bien	0.5	Bien		6.65
16636538	1.5	¡Perfecto! Deben indicar su pareja en el PDF	1.5		0.9	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 solo uno de la pareja para la próxima, pues habrán descuentos.	0.25	En una función con dominio y output acotado si se puede	0.5	Bien	0.5	Bien	0.25	Faltó regresión		6.40

16636600	1.5	¡Perfecto! Deben indicar su pareja en el PDF	1.5		0.9	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 solo uno de la pareja para la próxima, pues habrán descuentos.	0.5	Bien	0.5	Bien	0.5	Bien	0.25	Falta Regresión		6.65
16636732	1.5	¡Perfecto!	1	No separa los tests minimos para cada tipo de Coverage. No estan bien los tests en global (Hay uno extra). Deberia estar organizado en "test 1,2,3,4 para Statement coverage" "tests 1,2,3,4,5 para Branch" .. y asi	1	¡Sobresaliente! Muy buen uso del grafo con cada caso	0.25	En una función con dominio y output acotado si se puede	0.25	El costo monetario no es, solo la madurez	0.5	Bien	0.25	Enfoque de regresión		5.75
16636910	1.5	¡Perfecto!	1.5	Muy claro el grafo con colores!	1	¡Sobresaliente! Muy buen grafo	0.25	En una función con dominio y output acotado si se puede	0.25	No se habla de la madurez del proyecto	0.5	Bien	0.5	Bien		6.50
16637402	1.2	Si bien se aprecia el entregar muchos test por casos. Hacer un test que evalúan, por ejemplo, que retorne False porque no tiene dígitos, da el mismo beneficio que crear 4 inputs diferentes. Faltó incluir alguna justificación que diferencia lo que se busca testar con los 4 diferentes inputs.	1	Falta 1 test en Branch Coverage. Hay 2 tests extra en Condition Coverage	0.8	Hay un error en el análisis de conditional coverage (el alumno 2 del conjunto 1 si hace que tarea >= 4 evalue a false). En el caso de tu pseudo-código, ambos tienen 90% de conditional coverage (-0.2)	0.25	En una función con dominio y output acotado si se puede	0.25	La segunda parte es porque el proyecto no es largo	0.5	Bien	0.25	Fata regresión		5.25
16637593	1.5	¡Perfecto!	1	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 Branch" .. y asi	1	¡Muy bien!	0.5	Bien	0.5	Bien	0	Un mayor % de coverage no garantiza mayor calidad	0.25	Falta regresión		5.75
16637666	1.5	¡Perfecto!	1	No separa los tests minimos para cada tipo de Coverage. No estan bien los tests en global (Hay uno extra). Deberia estar organizado en "test 1,2,3,4 para Statement coverage" "tests 1,2,3,4,5 para Branch" .. y asi	1	¡Sobresaliente! Muy buen uso del grafo con cada caso	0.25	En una función con dominio y output acotado si se puede	0.25	El costo monetario no es, solo la madurez	0.5	Bien	0.25	Enfoque de regresión		5.75
16637860	1.25	Faltó el test donde se espera True cuando se cumplen todos los requisitos.	1.25	El test T1 esta demas para Statement Coverage. El nodo N1 no deberia estar en el grafo ya que la asignacion buyable= =false es la misma para todos.	1	¡Buena respuesta! Pero ojo con el caso 3 del segundo conjunto, dado que NP < 3.8, no se alcanza a evaluar AS. Esto no afecta el resultado final.	0.5	Bien	0.5	Bien	0.5	Bien	0.5	Bien		6.50
16637879	1.2	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 una justificación de que probarás casos bordes para llegar al True.	1.5		1	¡Muy bien!	0.25	En una función con dominio y output acotado si se puede	0.25	Nombra dinero	0.5	Bien	0.5	Bien		6.20
16638360	1.2	Si bien se aprecia el entregar muchos test por casos. Hacer un test que evalúan, por ejemplo, que retorne False porque no tiene dígitos, da el mismo beneficio que crear 4 inputs diferentes. Faltó incluir alguna justificación que diferencia lo que se busca testar con los 4 diferentes inputs.	1	Falta 1 test en Branch Coverage. Hay 2 tests extra en Condition Coverage	0.8	Hay un error en el análisis de conditional coverage (el alumno 2 del conjunto 1 si hace que tarea >= 4 evalue a false). En el caso de tu pseudo-código, ambos tienen 90% de conditional coverage (-0.2)	0.25	En una función con dominio y output acotado si se puede	0.25	La segunda parte es porque el proyecto no es largo	0.5	Bien	0.25	Fata regresión		5.25
16639103	1.5	¡Perfecto!	1.5		0.8	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 de cada cláusula, sino que todas sean al menos una vez true/false. No hay descuento pues no se les pasó como materia oficial.	0.5	Bien	0	Se habla de tiempo ejecución y no de madurez de proyecto en la segunda parte	0.5	Bien	0	No está bien enfocada la respuesta		5.80
16639332	1.25	Faltó el test donde se espera True cuando se cumplen todos los requisitos.	1.5		0.9	En general bien, pero ojo con la notación de grafos (-0.1), si bien se entiende las anotaciones de lineas, en rigor se deben representar bloques de código como nodos y control de flujo como aristas.	0.25	En una función con dominio y output acotado si se puede	0.25	El número de líneas está relacionado, pero no es el factor	0.5	Bien	0.25	Falta enfoque regresión		5.90
16639537	1.5	¡Perfecto!	1.5		1	¡Muy bien!	0.25	En una función con dominio y output acotado si se puede	0.5	Bien	0.5	Bien	0.25	De regresión		6.50
16640373	1.5	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 condición de contraseña como lo de menos de 4 dígitos consecutivos. De todas formas se entrega todo el puntaje porque se puede apreciar que cumple el objetivo de diseñar pruebas suficientes y mínimas (dado tu forma de responder) para el problema dado.	1	Condition Coverage y Branch coverage no tienen los tests minimos.	1	¡Muy bien!	0.25	En una función con dominio y output acotado si se puede	0.5	Bien	0.5	Bien	0.25	Enfoque regresión		6.00

16640888	1.5	¡Perfecto!	1	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 Branch" .. y asi	1	¡Muy bien!	0.5	Bien	0.5	Bien	0	Un mayor % de coverage no garantiza mayor calidad	0.25	Falta regresión		5.75
16641299	1.5	¡Perfecto!	1.5	Muy claro el grafo con colores!	1	¡Sobresaliente! Muy buen grafo	0.25	En una función con dominio y output acotado si se puede	0.25	No se habla de la madurez del proyecto	0.5	Bien	0.5	Bien		6.50
17636906																1.00
17636973	1.5	¡Perfecto!	1.25	Falta test donde no tiene suficiente dinero en Branch Coverage. No deberia estar solo en Condition Coverage	0.9	¡Muy bien en general! Pero ojo con la notación del grafo (-0.1), faltó el nodo final (return). Buena observación acerca de como varía el coverage dependiendo del pseudo código :).	0.25	En una función con dominio y output acotado si se puede	0.25	Falta madurez del proyecto	0.5	Bien	0.5	Bien		6.15
17637104	1.25	Faltó el test de que falte minuscula pero si tener mayuscula y número.	1.5		0.7	No es necesario un grafo específico para el condition coverage, el otro grafo que hiciste está bien. Buen análisis del branch coverage, pero el "al ojo" del conditional no es correcto (-0.3)	0.25	En una función con dominio y output acotado si se puede	0.5	Bien	0	El % de coverage no garantiza mayor calidad	0	Falta regresión		5.20
17637740	1.5	¡Perfecto!	1.25	Falta test donde no tiene suficiente dinero en Branch Coverage. No deberia estar solo en Condition Coverage	1	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 descuento porque es un tecnisismo). Hubiese estado 100% correcto si en lugar de asignar a aprueba, se retornaba true / false.	0.25	En una función con dominio y output acotado si se puede	0.25	Si bien la cantidad de funcionalidades esta correlacionado no es la viariable	0	No habla de que depende ya que un % de coverage mayor no garantiza mejor codigo	0.25	Falta sistema e integración		5.50
17637856	1.5	¡Perfecto!	1.5		1	¡Muy bien!	0.25	En una función con dominio y output acotado si se puede	0	Se habla de costos generales, la segunda relacionar con madurez de proyecto	0.5	Bien	0.25	Las más importantes son de sistemas e integración		6.00
17638488	1.2	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 una justificación de que probarás casos bordes para llegar al True.	1.5		1	¡Muy bien!	0.25	En una función con dominio y output acotado si se puede	0.25	Falta explicictar madurez del proyecto en primera parte	0.25	Falta explicitar que el % de coverage no garantiza calidad	0	Falta regresión sistema y/o integracion		5.45
17639182	1.5	¡Perfecto!	1.25	Falta test donde no tiene suficiente dinero en Branch Coverage. No deberia estar solo en Condition Coverage	1	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 descuento porque es un tecnisismo). Hubiese estado 100% correcto si en lugar de asignar a aprueba, se retornaba true / false.	0.25	En una función con dominio y output acotado si se puede	0.25	Si bien la cantidad de funcionalidades esta correlacionado no es la viariable	0	No habla de que depende ya que un % de coverage mayor no garantiza mejor codigo	0.25	Falta sistema e integración		5.50
17640121	1.5	¡Perfecto!	1.25	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 Branch!	0.8	Buen análisis, pero faltó mostrar el grafo asociado (-0.2).	0.25	En una función con dominio y outputs acotado si se puede	0.5	Bien	0.5	Bien	0.25	Faltó mas precisión en la respuesta		6.05
17640407	1.5	¡Perfecto!	1.25	Falta test donde no tiene suficiente dinero en Branch Coverage. No deberia estar solo en Condition Coverage	0.9	¡Muy bien en general! Pero ojo con la notación del grafo (-0.1), faltó el nodo final (return). Buena observación acerca de como varía el coverage dependiendo del pseudo código :).	0.25	En una función con dominio y output acotado si se puede	0.25	Falta madurez del proyecto	0.5	Bien	0.5	Bien		6.15
17640598	1.5	¡Perfecto!	1.25	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 Branch!	0.8	Buen análisis, pero faltó mostrar el grafo asociado (-0.2).	0.25	En una función con dominio y outputs acotado si se puede	0.5	Bien	0.5	Bien	0.25	Faltó mas precisión en la respuesta		6.05
20405332	1.5	¡Perfecto!	1.5		1	¡Muy bien!	0.25	En una función con dominio y output acotado si se puede	0.5	Bien	0.25	Hay más aspectos a considerar	0.5	Bien		6.50
1562188J	1.5	¡Perfecto!	1.25	Hay un test extra en Condition Coverage. Deberian ser 6	0.9	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 booleano sea evaluado como true o false por si solo, no en base al predicado completo.	0.25	En una función con dominio y output acotado si se puede	0.5	Bien	0.5	Bien	0.25	Falto decir explicitamente regresión		6.15
1563339J	1.5	¡Perfecto!	1	Branch Coverage y Condition Coverage no tienen el numero minimo de tests. Deberian ser 4 y 6 respectivamente	0.5	Faltó el 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 >= 0.9), por lo que esos casos que mencionan que no se cubren no aplican para el análisis (falla completamente en análisis de branch coverage, -0.3)	0.25	En una función con dominio y output acotado si se puede	0.5	Bien	0.5	Bien	0.5	Bien		5.75
1563373J	1.5	¡Perfecto!	1	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 Branch" .. y asi	1	¡Muy bien!	0.25	En una función con dominio y output acotado si se puede	0.25	Primera parte hablas de costo, demasiado general	0	Un mayor % de coverage no garantiza mayor calidad	0.25	No nombra regresión		5.25

1563390J	1.2	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 una justificación de que probarás casos bordes para llegar al True.	1.25	Hay un test extra en Condition Coverage. Deberian ser 6 tests para esa cobertura.	0.8	Bien en general, pero faltó el grafo (-0.2). Ojo 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 es un concepto que no abordamos en clases y tu interpretación puede estar correcta.	0.25	En una función con dominio y output acotado si se puede	0.5	Bien	0.5	Bien	0.25	El de usuario ya debería haber sido testeado de antes	2	3.75
1563759J	1.25	Faltó el test donde se espera True cuando se cumplen todos los requisitos.	1	El test 1 no deberia estar en Statement Coverage. Hay 2 tests que son extra en Condition Coverage, deberia ser el minimo (6 tests)	1	¡Muy bien! Eso si, el conditional coverage pide que cada statement booleano por si solo evalue a true y false, no en conjunto. No se descuenta porque la fuente puede ser válida igual.	0.25	En una función con dominio y output acotado si se puede	0.5	Bien	0.5	Bien	0.25	Enfoque de regresión		5.75
1563941J	1.5	¡Perfecto!	1.25	Hay un test extra en Condition Coverage. Deberian ser 6 tests para esa cobertura.	1	¡Bien! Pero ojo con los conceptos para el branch coverage. La rama es de un nodo a otro, lo que mencionaron en su respuesta son caminos.	0.5	Bien	0.5	Bien	0.5	Bien	0.25	Falta regresión		6.50
1663523J	1.5	¡Perfecto!	1.25	Falta un test en Condition Coverage: Tiene dinero suficiente, no es recomendado, es el volumen 1 y no tiene más de 7 volúmenes.	0.8	Buen código, análisis y presentación de los casos, pero se pedía justificación con grafo (-0.2).	0.5	Bien	0.5	Bien	0.5	Bien	0.25	Falta integración y sistemas		6.30
1663649J	1.5	¡Perfecto!	0.75	Revisar las definiciones de cada tipo de coverage. Hay un tests extra en condition coverage, deberia ser el minimo numero de pruebas. No pueden haber mas tests minimos en statement coverage que en branch coverage. Uno es "subconjunto" de otro.(Statement <= Branch)	0.9	¡Bien en general! Pero ojo con la notación del grafo (-0.1), las condiciones se evalúan como un único nodo a pesar de tener múltiples statements booleanos, que simplifica bastante el análisis.	0.5	Bien	0.5	Bien	0.5	Bien	0.25	El enfoque más importante es de regresión		5.90
1664171J	1.25	Faltó el test donde se espera True cuando se cumplen todos los requisitos.	1.25	El test T1 esta demas para Statement Coverage. El nodo N1 no deberia estar en el grafo ya que la asignacion buyable= =false es la misma para todos.	1	¡Buena respuesta! Pero ojo con el caso 3 del segundo conjunto, dado que NP < 3.8, no se alcanza a evaluar AS. Esto no afecta el resultado final.	0.5	Bien	0.5	Bien	0.5	Bien	0.5	Bien		6.50
2010233J	1.3	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 consecutivos).	0.5	Ese conjunto de datos no cumple con todas los Coverage. Deberian ser 3 para Statement Coverage, 4 para Branch Coverage y 6 para Condition Coverage	0.5	Falto el pseudocódigo (-0.2), inferí para la corrección que era un if/elsif con asignación dentro de los bloques. El análisis de condition coverage es incorrecto (-0.3), se debe evaluar cuántos casos hay con cada condición en true/false y ver cuántas cumple cada conjunto.	0.25	En una función con dominio y output acotado si se puede	0	Se habla de costo general, el costo es la consecuencia	0.5	Bien	0	Fala enfoque regresión y sistema y/o integración		4.05