

Grupo	Último commit a master	P1 Correcciones comentarios entrega 1	Comentarios P1	P2 Cumplir con todos los puntos de los requisitos descritos en la sección 3	Comentarios P2	P3 Desarrollo de funcionalidades	Comentarios P3	P4 Tests	Comentarios P4	Nota
1	https://github.com/ITC3745-2020-2ygrupo01commits/39d4c51ae4e17e58d3b7946c0d097b760b0a9d	Destacado	Bien, incluyen link a heroku y mejoran commit	Destacado	Git / gitflow - buen gitflow. Mejoraron bastante los commits. Sería bueno que todos usaran la convención de "fixcoop message", pero no es mandatorio. Configuraciones - todos los chequeos tienen excelentes métricas.	README - bien	No se si modificaron su planificación (hicieron algo de préstamos/deudas pero eso estaba para el sprint 3). No obstante, las funcionalidades se ven bien y tienen una carga similar. Bien en ese sentido. A la única funcionalidad que le falta algo es a la de transacción, que debiera tener un select para el "account_id" en lugar de un campo de texto. Sin embargo, tengo un vacío importante respecto a cómo plantearon las deudas y préstamos: dado que estamos hablando de plata real, si yo presto 1000 esto debe salir de una cuenta (por ejemplo, de mi cuenta corriente). Por otro lado, cuándo pago una deuda esto debería llegar a alguna de mis cuentas (por ejemplo, cuenta corriente). ¿Cómo se está viendo esto en la plataforma? De momento entiendo que las bandejas pertenecen al usuario, pero en rigor deberían pertenecer a una cuenta. Revisen esta issue si tienen dudas acerca de la implementación: https://github.com/ITC3745-2020-2yylab/issues45 . Definir bien y corregir esto para la próxima entrega.	Destacado	Excelentes sus tests, los controladores testearon bien los cambios en base de datos, hicieron tests de sus helpers, rutas y vistas. Buenos tests de modelo y buen uso de should-matchers. Las únicas dos observaciones que les recomiendo fuertemente son: - Hacer un stub o mock de URI para la llamada de su scraper a la conversión de moneda. Este test falla si corto el internet y no significa que el código esté mal, sino que a mí me falta algo. - Agregar tests unitarios de los helper methods más importantes de sus controladores de transacciones y validateds. Si se les complica mucho hacerlos así, pueden mover esa lógica a los modelos y testearlos desde ahí.	6.7
2	https://github.com/ITC3745-2020-2ygrupo02commits/39d4c51ae4e17e58d3b7946c0d097b67d157a53	Destacado	Bien el cambio en la exigencia del coverage. Muy bueno el deploy a dos ambientes, excelente práctica.	Destacado	Git / gitflow - buen gitflow y muy buenos commits, que bueno que todos hayan adoptado convencional. Configuraciones - todos los chequeos de calidad corren bien. Excelente. Las reglas ignoradas de rubocop tienen sentido y/o comentario asociado (bastante más).	README - Perfecto	Buena decisión de framework y add-on para el diseño. Mucho más de lo esperado para el curso. Algunas observaciones que no requieren arreglo pero por si lo quieren perfeccionar: - Los selects están un poco bugeados eso si (a veces desaparecen y aparecen o el anchor de los items se mueve hacia abajo) - El perfil de cuenta tiene un edit abajo y además el ícono de herramienta. - Las cuentas en cuyo postion ocultan el input. Veo que en el edit no lo muestran por ejemplo. Todas las funcionalidades planeadas se ven bien ejecutadas y funcionales. Buen trabajo.	Más allá de lo esperado	- Si bien usan scaffold, en los tests de controlador haría un chequeo de que hubo un cambio en la base de datos al hacer alguna acción (+1 entrada al crear, -1 entrada al borrar, verificar el atributo editado al update). Eventualmente puede modificarse esa lógica y no habrá tests que le haga guardia. - Fallaron tests de rutas de al menos las Accounts. No lo olviden para las próximas entregas. (no descuento ahora, pero si para la próxima) - Fallaron tests unitarios de los métodos de Account (, no les quiero bajar pero son varios. La idea es tenerlos controlados). - Deberían testear los helpers también (en este caso solamente de Account). No hay descuento porque no se explicita en el enunciado, pero de ahora en adelante considerémoslo.	6.7
3	https://github.com/ITC3745-2020-2ygrupo03commits/39d4c51ae4e17e58d3b7946c0d097b67d157a53	Bueno	No hay protección para las rutas, puedo iniciar sesión como un usuario y editar los datos del otro (cambiando el atributo action del form). Ojo con esto, era parte de la primera entrega. Corregir si o si para la próxima (Pundi es fe bueno en este ámbito). Por otro lado, no volvieron a subir el E/R.	Bueno	Git / gitflow - Comentarios similares de la entrega pasada, buen uso de gitflow y commits porciemente buenos. Recuerden, detallen un poco en los mensajes de sus commits y traten siempre de hacer commits pequeños (-) (por ejemplo, el both 118 es enemy y asumo que fue un error, 0479d6d se pudo haber separado bastante más). Configuraciones - Rubocop tiene 62 ofensas de las cuales 36 son autocorregibles (con rubocop -a). Ojo con eso, es un comando + un commit y ya les baja la mitad. - esint tiene 3 errores - erubini tiene 26 errores - el resto de los lint bien - bundler audit dice que tienen vulnerabilidades (lectopack) - brakeman bien	README - Buen README, incluye link a Heroku y las observaciones para la conexión	Por los commits y su README, veo que empezaron bien encima la entrega. Ojalá corrijan esto, es poco lo que se pueden enfocar en calidad si están siempre contra el tiempo. Dado que uno de sus compañeros botó el ramo, considerare menos funcionalidades (5 de su planificación actual) en total (los dejo a su criterio, pero actualicen la planificación). Comentarios: 1. En los tipos de cuenta se empieza con un balance de 100, corregir esto. A la creación le falta poder asignar un número de cuenta. No hay update ni delete. 2. No se pueden crear transacciones porque tienen un error en la migración: el "account_id" no puede ser false y si debiera poder serlo cuando el tipo de transacción es income o expense. 3. Se pueden transferir montos negativos. Corregir 4. Las categorías se pueden crear pero no editar ni borrar, las páginas quedaron con las vistas por defecto (al menos hasta el commit antes de la fecha límite). Corregir 5. Las categorías son manejables por un usuario no admin. Mejorar para la próxima entrega. 6. La navegación no cumple, tiene que ingresar a la mayoría de las rutas vía URL y buscando en el routes.rb cuales estaban disponibles. Agregar botones de navegación para la próxima	Correcto	Controladores: crear un account/transaccion/category y verificar que esta se añada a la base de datos, lo mismo con edición y borrar. La response no es suficiente para asegurar el funcionamiento. - Modelos: estos están casi perfectos, agregaría un test para validar el RegEx del email. Ojo que si agregan métodos tienen que agregar tests unitarios de ellos. - Rutas: fallaron las rutas de los nuevos modelos Account, Transaction y Category - Vistas: verificar que un template con input controlado sea consistente con lo que bucaron diseñar (e.g. que muestre el número de cuenta o el nombre de usuario) Ojo con esto, los de balance son más complejos todavía y después vienen tests más complejos de vista con capybara. No dejen todo para última hora.	4.25
4	https://github.com/ITC3745-2020-2ygrupo04commits/0a5ed6e1390f296fcae20da14ae7a8736f10ac2a6	May bueno	Sobre las correcciones de seguridad a las páginas indicadas, se corrigieron. No obstante, tuvieron otras páginas que no tienen seguridad, pero eso lo detallaré en otra ítem. Por otro lado, se les comentó que el proyecto exige usar Should-Matcher y todavía no se ocupa en los tests. Tengo que descontar aquí un poco porque no corrigien ese comentario con los test antiguos que habían creado.	Más allá de lo esperado	1. Respecto a los aspectos técnicos. Todo bien en el gitflow y configuración de archivos. No obstante, todavía sus commits no son descriptivos. Commits como "rubocop" ¿qué significan? ¿qué cambio significa eso? ¿actualizaron configuración de rubocop o actualizaron activarse porque rubocop indica que no siguen el formato? Para la siguiente entrega hagan commits más explicativos que den a entender qué cambio implica ese commit. 2. Sobre las configuraciones, lo único que falta incluir es el uso de Should-Matcher que en el enunciado se exige ocupar y todavía no hay test sobre eso. Recomiendo revisar el ejemplo creado por el profesor para que veien como utilizarlo en sus tests. Para los tests que tienen, pueden dejarlo así, pero todos los nuevos test que incluyan, se espera que sean con Should-Matcher como pide el enunciado. Extra sin descuento: Github también indica que tienen vulnerabilidades, deben revisarlas para la entrega 3.	Cumple con lo esperado	1. En términos de funcionalidades, están todas las pedidas, pero la página no está alineada a las funcionalidades, es decir, para algunas la página está en español y con un form adecuado y en otras está en inglés. 2. Las categorías aparecen como un número en la tabla de resumen en vez de la palabra esperada y también se pueden crear 2 con el mismo nombre. Si las categorías solo tienen un nombre, se esperaría que fueran únicas. 3. Los préstamos y deudas no está completo. Un usuario debe poder saber distinguir una deuda que luego debe pagar o un préstamo que luego se lo deberán devolver VS un gasto o ingreso que ya es una transacción finalizada. Imaginen que deuda tiene 2 tests. Una donde indicas que tiene la deuda (pero tu saldo no es afectado, solo el no liquidado) y otro donde indicas que la deuda fue saldada y ahí se disminuye tu saldo. Revisen la issue 45 https://github.com/ITC3745-2020-2yylab/issues45 4. La funcionalidad de usuario no está totalmente finalizada. Una requerimiento de un usuario es que con dicha credencial puede acceder a sus datos y nadie más puede, pero yo como usuario si me puedo meter a la cuenta de otro https://testing-group04.herokuapp.com/accounts/2 y realizar acciones. 5. No se debería permitir ingresos negativos que se transforman en gastos o gastos negativos que se transforman en ingresos. 6. Las transferencias pueden ser negativas, situación que no debería pasar porque implica que le estoy quitando plata a alguien. 7. Las vistas de administrador deberían incluir una forma de volver atrás y finalizar cas). 8. Piensen en qué sucede si elimino una categoría o divisa, ¿qué sucede con en la tabla de ingresos por ejemplo con los ingresos de dicha categoría? 9. La planificación no se ajusta totalmente a las funcionalidades presentes. Por ejemplo, la página permite deshabilitar a un usuario (funcionalidad de la entrega 1) y por lo tanto, intento probarla y se ve que si logro transferir dinero a pesar de estar inhabilitado. Eso implica que la funcionalidad está incompleta. Se recomiendo no adelantar trabajo de una entrega si estará incompleta, para eso mejor dejarlo en una branch aparte y hacer merge post entrega 2. No se descontó por esta funcionalidad en específico, puesto que el compromiso era tenerlo en la entrega 3. 10. Se esperaba un documento con la planificación actualizada de que trabajo iba a ser en cada sprint (en especial los que se vienen dando que adelantaron algunas cosas). No está dicho documento. Me basé en el documento de la primera entrega + el comentario que les había puesto. Para la entrega 3 deben actualizar la planificación. Se espera que para la entrega 3 corrijan todos estos comentarios y si quieren adelantar funcionalidades de la entrega 4, deben hacerla en su completitud (para poder evaluar) o no las corrijan y así no evaluar dicha funcionalidad. De este modo me enfoco de evaluar lo que realmente se espera de dicha entrega. Finalmente, recomiendo que luego de arreglar estas cosas, reinicien la base de datos y ligan un seed con cosas ya creadas.	Cumple con lo esperado	1. En los test de modelo solo validan valores posibles, lo cual estaba bien para la entrega 1 que solo tenían 1 modelo, pero hay muchas cosas más que testear en un modelo, tales como: associations, uniqueness, attributes (revisar algo por example que presento el profesor en clases). Esos test se vuelven muy relevantes ahora que tienen una gran cantidad de tablas y relaciones entre ellas. Agregarlos para la entrega 3. 2. Sobre los test del controlador, si bien aseguran que la app funciona correctamente, los test sirven para más que eso, también para validar la calidad de la aplicación, en torno a asegurar que las acciones que haga un usuario sean válidas o no tales como prestamos negativos, acceso a rutas que no corresponde, si indica que va a transferir entre sus cuentas, que no le acepte transferir a otra cuenta, etc. Piensen en qué acciones no son deseadas de un usuario y esas deberían estar testeadas para asegurar que ante cualquier cambio, esas acciones siguen siendo rechazadas. Hasta ahora testean demasiados happy paths, lo cual está super, pero falta pensar en que los otros caminos también sean testados. 3. Sobre test de rutas, todo bien. 4. Sobre test de vistas, no hay ninguno que revise que el contenido de los templates sea consistente con la información del backend. No obstante, se dejará pasar porque en la siguiente entrega deberán usar Capybara que será justo para validar las vistas.	4.7

Grupo	Último commit a master	P1 Correcciones comentarios entrega 1	Comentarios P1	P2 Cumplir con todos los puntos de los requisitos descritos en la sección 3	Comentarios P2	P3 Desarrollo de funcionalidades	Comentarios P3	P4 Tests	Comentarios P4	Nota
							1. Una cuenta de debito no puede tener saldo negativo y en este caso, si pude indicando un gasto cuando tenia saldo 0. 2. El usuario puede acceder a la info de las cuentas de otro usuario. Eso iría en contra de lo que se espera de un usuario que solo ve su información. o a lo más el admin debería poder ver lo demás. 3. Tal como en gasto/ingreso explican que significa el negativo. En "Crear una Préstamos o deudas" deben explicar también que implica el negativo, si pongo -5000, ¿luego una deuda de 5000 que acabo de pagar y por eso perdí 5000? ¿hice un préstamo de 5000 y por eso pierdo 5000 de mi cuenta, pero luego lo recupero? Por otro lado, eso influye directamente en el saldo, pero un préstamo y deuda no afectan directamente el saldo (como lo hace un gasto/ingreso). Recomendó revisar la issue 45 https://github.com/ITC3745-2020-2/yvylab/issues/45 , porque prestamos y deudas tiene 2 fases, una donde declaras el préstamo/deuda y otro donde hace efecto en el saldo de tu cuenta. 4. También, no hay diferencia en interfaz entre un prestamos liquidado o no liquidado, ambos afectan el saldo que se muestra. 5. En las transacciones no se diferencia entre gastos/ingreso y préstamos/deuda, podrían agregar una columna en la tabla para distinguir eso. 6. En la tabla de detalle, sale que un ingreso no está liquidado, cuando este ya debería estar liquidado, puesto que afecta directamente el saldo de tu cuenta, no es como un préstamo que luego te llegará el dinero. 7. Se puede hacer transferencias negativas para ganar dinero. 8. En la tabla de "Últimos movimientos" > A mis cuentas", la columna de "Cuenta" sale siempre vacía. 9. Podrían distinguir en la tabla una transferencia VS un ingreso o gasto. Al final, agregar una columna que diga el tipo de la transacción. 10. En la vista de admin, se cae cuando pongo que quiero ver las transacciones. Si un admin no tiene permiso para ver eso, no debería aparecer dicha opción en la barra superior. En caso de si tener permiso, algo está mal que genera un error. Recomendación: Cuando te ofrece el destinatario de cuenta, podrían poner "usuario-cuenta" para que sea más fácil saber a qué cuenta transferir. Esto lo digo porque me hice 2 cuentas que llamé mi cuenta "Debito" y luego confundí cual "Debito" es de cual usuario. Dado que en la planificación sale que todo el tema de saldo liquidado y no liquidado lo verán en la entrega 3 (punto 3, 4, 5 y 6), no se aplicará descuento por los puntos que irán de la entrega 3, pero se los hago notar para que no lo dejen de lado y revisen que todos mis comentarios queden resueltos con la entrega 3.			
5	https://github.com/ITC3745-2020-2/grupo05/commits/041a59061f7d90626d87e4c7b9eb1262072d5b	Destacado	Muy bien en cambiar los comentarios indicados.	Destacado	Extra sin descuento: Github también indica que tienen vulnerabilidades, deben revisarlas para la entrega 3.	Muy bueno		Más allá de lo esperado		6.25
6	https://github.com/ITC3745-2020-2/grupo06/commits/3d59b3a9c2ae3957defd2a47ef8f9dc07d6c9a8f	Más allá de lo esperado	Se comentó que faltaron test de vistas y no lo agregaron para esta entrega	Muy bueno	1. Excelente con el giflow D 2. Falto indicar en el readme las credenciales para un admin, puesto que en esta entrega hicieron funcionalidades de admin que debía revisar pero no es posible sin credenciales. 3. Brakenheim indica que tienen una vulnerabilidad, deben solucionarla para la otra entrega. Por otro lado, rubocop también empuja demasiadas alertas, deben revisar dichas alertas para analizar cuales deben arreglar y cuales podrían omitir (actualizando el archivo de configuración de rubocop) de este modo, todos los test deben pasar sin indicar un error. 4. Para esta entrega que iban a tener muchas más funcionalidades y test, se esperaba que algunos (ideal si fueran todos) fueran como se indica en el enunciado con Should-Matcher. Para la siguiente entrega, se espera que los test de controlador que implementen, sean con esta gema. Extra sin descuento: Github también indica que tienen vulnerabilidades, deben revisarlas para la entrega 3.	Correcto	Cumple con lo esperado		4.85	
7	https://github.com/ITC3745-2020-2/grupo07/commits/3d483ac071907f7c1d0c1522484d6d3a0c5059	Más allá de lo esperado	No agregan las instrucciones de como correr los tests. Bien por agregar docker en el readme	Destacado	1. Giflow: Bien, sin comentarios 2. Configuraciones: Todo ok	Insuficiente		Correcto		4.85
8	https://github.com/ITC3745-2020-2/grupo08/commits/a55626a7a9a9d88339e413a90c1c0d6247f9d82c	Destacado	No hay instrucciones de como correr la aplicacion o tests en README	Más allá de lo esperado	1. Giflow: Todo ok 2. Bien configuraciones. https://github.com/ITC3745-2020-2/grupo08/network/alert/ryan.lock/node-forgelopen Tienen que actualizar las dependencias que tienen vulnerabilidades	Bueno	- No hay validaciones en montos de ingresos/expenses y tampoco al hacer transferencias. - No tiene sentido que deba elegir los IDS de las cuentas para poder transferir. Funcionalidad la considero incompleta por esto. - Validación de registro no me muestra el campo incorrecto de email (intente de hacer una cuenta con el mail las@valleltest.cl) - El label de Email está mal colocado o por lo menos fue confuso de crear una nueva cuenta. - Cuenta de debito no puede tener deuda - No debería aparecer el ID de la cuenta origen como opcion de destino (No deberían ser IDS tampoco)	Bueno		5.6
9	https://github.com/ITC3745-2020-2/grupo10/commits/3d59b3a9c2ae3957defd2a47ef8f9dc07d6c9a8f	Muy bueno	Todo bien con los comentarios de la entrega pasada	Muy bueno	1. Giflow: ¡Bien! 2. Rubocop no está pasando. Tienen vulnerabilidades que deben arreglar en Github	Más allá de lo esperado	- Solo implementan CLP. - Casi todo esta en crudo HTML y habían realizado un MockUp bastante detallado. No es el propósito del curso demandar CSS profesional pero pueden usar Bootstrap o otras librerías para organizarlo en unos minutos. - No tiene mucho sentido por editar las cantidades de los Accounts. No representa un flujo real de la logica de negocio. (La única excepción de logica de negocio es la cantidad inicial por simplicidad del problema, pero no poder editar esa cantidad después). - Faltan validaciones al hacer transferencias. Mostrar mas informacion de las cuentas que los IDS ("No se que cuenta es cual") - No validan que no se ingresen decimales con varios lugares ("0.0000000000000") se cae la aplicación. Deben cubrir los casos bordes en los tests - Edit de transacciones no debería ser accesible y se cae al entrar a esa ruta (Desde la lista de transacciones). No debería poder eliminar una transacción	Más allá de lo esperado		5.8
10	https://github.com/ITC3745-2020-2/grupo10/commits/3d59b3a9c2ae3957defd2a47ef8f9dc07d6c9a8f	Insuficiente	El link de heroku no hace nada. Solo crear accounts	Bueno	- Aplicacion no está montada en Heroku - Vulnerabilidades de dependencias en Github que deben arreglar - Rubocop no está pasando	Muy insuficiente	- Esta bien que el enfoque del curso no sea en desarrollar vistas con CSS profesional pero hay varias vistas donde no puedo interactuar porque los botones están muy cerca y no logro distinguirlos. La idea es mantener la usabilidad mínima pero casi no logro usar las funcionalidades. - El Show de transacciones no funciona! Edit se cae. - Vista de "Debito" que me redirecciona después de borrar una deuda no funciona. Se cae al crear un nuevo "Debit". Esto es lo que pasa por usar Scaffolds. Lo mismo con Income y los otros tipos de Transacciones. - Settled: Balance no hace nada. No muestran balance de una cuenta, siempre es "0" - No validan monto negativos al hacer transacciones	Insuficiente		3.1

Grupo	Último commit a master	P1 Correcciones comentarios entrega 1	P2 Comentarios P1	P3 Comentarios P2	P4 Comentarios P3	P5 Comentarios P4	Nota		
11	https://github.com/IC-3745-2020-2/grupo11commits/3a85d1e1158a23070d75ec11377aef702bed4e2	Muy bueno	Les dejé comentado que los tests era lo más importante pero como veían más adelante dejaron un poco botado los tests de modelos	1. Me salieron 202 ofensas en rubocop 2. Los lint me fallaron 3. Buenos commit 4. Bien montada la app	1. Van bien con las vistas, pero no puede haber texto en inglés y español preocuparse que los inputs estén alineados. Al momento de realizar una transferencia no se actualiza el saldo líquido. La foto de perfil no se ve. 2. Las funcionalidades cumplen en el happy path, pero fallaron muchas validaciones -> puede realizar un ingreso negativo -> esto mismo fallo agregar a los tests también. 3. En general la idea de la aplicación era no hacerla demasiado extensa justamente para que se dedicaran a medir cada validación y luego lo idealen en los test	Correcto	Correcto	1. Los tests de modelos están bastante pobres, por ejemplo para las cuentas podrían probar que efectivamente se tira un error cuando no está el atributo currency. También podrían probar que una cuenta debe pertenecer a un usuario. 2. Los tests de controladores están bastante bien. 3. Para la siguiente entrega les voy a asignar algunos tests de vistas	5
12	https://github.com/IC-3745-2020-2/grupo12commits/a514e85d1e8a72945a34d89d43400b2d12d5d2d	Muy bueno	- Se agregaron las transferencias. - No se agregaron divisas. - No se han agregado pagos en cuotas - Se agregó una navegación - No se protejeron las vistas	Más allá de lo esperado	- La aplicación se cae si la cuenta tiene saldo == 1,000.000 - La aplicación se cae en transferencias a cuentas que no existen en La plataforma. - Me pudo transferir a mi propia cuenta y se crea plata - Deben proteger las vistas de admin (como usuario no admin puedo ver todos los usuarios, las cuentas, movimientos, deudas etc de todos) - Agregaron estilo a la aplicación y se ve bastante bien - Respetaron su planificación (menos el agregar divisas a las cuentas)	Muy bueno	Muy bueno	- rspec: tests pasando coverage: 100% - bien el uso de shoulda-matchers en tests de modelos y rutas, les recomiendo verlos para los controladores también - tests modelos: están bien pero también es importante testear otras cosas como asociaciones, validaciones, etc. - tests controladores: sus tests de controlador revisan que las acciones básicas del crud funcionen, pero no qué sean correctos. - Venen los otros comentarios del feedback, hay hartas cosas que podrían darse cuenta si los tests fueran de mejor calidad. - bien los tests de vistas	5.75
13	https://github.com/IC-3745-2020-2/grupo13commits/215a5d5b999a39a9702a6b5d452a6182102e63e	Bueno	Los comentarios que agregaron en el readme el comando para correr rubocop de manera automática y que estaba poco claro -> no lo editaron	Insuficiente	1. Si bien no se espera una vista hermosa de diseño si se espera que no sea el default 2. El flujo está muy pobre me quedaba atorado sin poder volver al "menu" 3. No pude realizar prácticamente ninguna funcionalidad de la planificación solo crear cuentas a pesar de que algo vi en los controladores	Muy insuficiente	Insuficiente	Veo que realizaron algunos tests pero casi todos los errores, espero poder ver para la próxima entrega un poco más de dedicación. Revisé que no realizaron aversos hasta el último día	3.1
14	https://github.com/IC-3745-2020-2/grupo14commits/215a5d5b999a39a9702a6b5d452a6182102e63e	Destacado	Hicieron lo pedido	Destacado	1. Cuando quiero realizar una transacción estando en una cuenta luego debo volver a seleccionar la cuenta 2. Se que tienen funcionalidades para el proximo sprint, pero como feedback de antemano asegurarse que el flujo y las diferencias entre débito, crédito se puedan ver de manera intuitiva 3. Manejar que solo haya un idioma inglés o español 4. El rol eliminado de la creación de usuario -> creen uno y coloquen en el readme	Más allá de lo esperado	Destacado	Muy completos los tests los felicito	6.7
15	https://github.com/IC-3745-2020-2/grupo15commits/2004c5ee9e1f5d8bbe9bdc325c226d4b58e1c0d3	Más allá de lo esperado	Agregaron casi por mínimo - Mejoraron sus prácticas de gitflow	Muy bueno	- gitflow: correcto, les recomiendo ir borrando las branches que ya fueron mergeadas para ordenar un poco - rubocop: ok tienen hartas cosas ignoradas en los archivos, intenten solucionarlas más que ignorarlas, aplica para todos los ítems vi una cosa de sass-lint ignoradas también. Busquen las reglas, muchas veces tienen recomendaciones de como se debería hacer, se aprende harlo de eso y se mejora la calidad del código por algo están los ítems...) - eslint: ok - erb-lint: ok - brakeman: ok - heroku / github actions: se cae con el signpur-login. No actualizado	Bueno	Bueno	- rspec: tests pasando coverage: 100% (El coverage no lo es todo en los tests) - tests modelos: bien los tests de modelo, muy completos! - tests controladores: mejoraron la exhaustividad de los tests pero todavía falta, pónganse en casos reales por ejemplo casos que deberían fallar y asegúrense que fallan, intenten que los tests realmente tengan valor (pueden jugarlos por los otros comentarios) que les dejen, son cosas que deberían testearse! - bien los tests de rutas y vistas! - Shoulda-matchers, no los están usando y son muy útiles incluso para saber que testear, investiguen sobre ellos y úsenlos para la próxima entrega!	5.45
16	https://github.com/IC-3745-2020-2/grupo16commits/2004c5ee9e1f5d8bbe9bdc325c226d4b58e1c0d3	Muy bueno	- Bien el cambio y orden de las historias de usuario - Bien que agregaron la planificación de sprints al trello - Bien las mejoras al modelo de datos - Todavía tienen muchos temas de privacidad/seguridad incluso sin estar loggeado - Falta css	Insuficiente	- Las divisas hoy no afectan en nada, si transfero 3 a una cuenta en usd se suman 3 al saldo solamente - En raro que como admin pueda hacer transacciones de usuarios - El botón de logout está solamente en la vista de usuarios - Deben agregar css a la página, no tiene que ser hermosa pero no pueden quedarse sólo con las vistas base - Edit account no funciona (probado como usuario no admin) - Se está outofrole al acceso a las vistas pero no se están bloqueando (puedo entrar por ruta a editar un usuario que no voy yo a ver las cuentas de otro usuario, etc) - Sólo se puede transferir a la cuenta 1 (debe estar seteada por default, no toma la cuenta que yo le pongo como target) - No se transfiere si no tengo saldo (no se crea plata) pero no se me avisa que la transacción no se hizo - Las cuentas no tienen tipo	Bueno	Insuficiente	- rspec: hay 4 tests fallando coverage: 26.32% - tests modelos: No hay que el coverage marca que están pasando otro código pasar por ahí pero no se están testando los modelos, sus atributos, sus restricciones, sus asociaciones, etc. - tests controladores: litan y falta exhaustividad, casos en que fallan, casos bonides, etc. - No hay tests de rutas - Ok tests de vistas, para las próximas entregas será más relevante esto especialmente con capybara	4.1
17	https://github.com/IC-3745-2020-2/grupo17commits/04434701198f1a887a6c127be96e27c139a95d9	Destacado	Destacado	Destacado	1. Faltan algunas validaciones o feedback al usuario, por ejemplo, podía cobzarte un monto máximo a la dea de debito negativo. También cuando intento transferir plata negativa no me dejó pero no me explico la razón. 2. Creo que estaría bueno mejorar un poco las vista van bien pero faltan unos margenes o paddinga que no se demorar más de 10 minutos en cargar	Más allá de lo esperado	Destacado	Van bien, eso si empezaría a trabajar desde ya en la siguiente entrega porque la veo pesadita	6.7
18	https://github.com/IC-3745-2020-2/grupo18commits/2004c5ee9e1f5d8bbe9bdc325c226d4b58e1c0d3	Bueno	- Aún no han corregido los temas de seguridad/privacidad todas las acciones de CRUD pueden hacerse incluso sin estar loggeado, algunos links están exconorde pero accediendo por ruta no hay ninguna vista bloqueada. - Falta css - Mejoraron sus prácticas de gitflow	Bueno	- gitflow: está bien, recuerden que lo que se revisa es lo que está en master, deben hacer las PR antes de la entrega tienen branches con muchos cambios que no los hicieron merge - rubocop: ok - eslint: falta - sass-lint: muchas ofensas - erb-lint: no come - bundle-audit: ok - brakeman: ok - heroku / github actions: deploy correcto y automatizado - README: agregar credenciales de usuarios de local	Bueno	Muy bueno	- rspec: tests pasando coverage: 100% - tests modelos: están bien en general pero hay más cosas que se pueden testear como asociaciones entre modelos o restricciones de los atributos - tests controladores: bien, bastante completos y exhaustivos. todavía se puede mejorar por ejemplo en los temas de seguridad que les menciono - tests de vistas y rutas: bien pero faltan - Shoulda-matchers, no los están usando y son muy útiles incluso para saber que testear, investiguen sobre ellos y úsenlos para la próxima entrega	6.15