

Universidad ORT Uruguay

Facultad de Ingeniería

Bernard Wand Polak

Diseño de Aplicaciones 1

Obligatorio 1

https://github.com/IngSoft-DA1-2023-2/187884_303433_303804

Emiliano Marotta – 187884

Sebastian Borjas – 303433

Agustina Martinez – 303804

ÍNDICE:

INTRODUCCIÓN:.....	3
DISEÑO Y DIAGRAMAS:.....	3
PRINCIPALES DECISIONES TOMADAS:.....	7
COBERTURA Y FUNCIONALIDADES:.....	8
Link de video de Youtube:.....	9

INTRODUCCIÓN:

El sistema fue implementado siguiendo las exigencias del proyecto. Por una cuestión de tiempo decidimos centrarnos en crear algo funcional y la documentación, por lo que quedaron las siguientes funcionalidades sin implementar.

Funcionalidades restantes en el backend:

- Eliminación de un usuario invitado de un workspace.
- Muestra de reportes.

Funcionalidades restantes en el frontend:

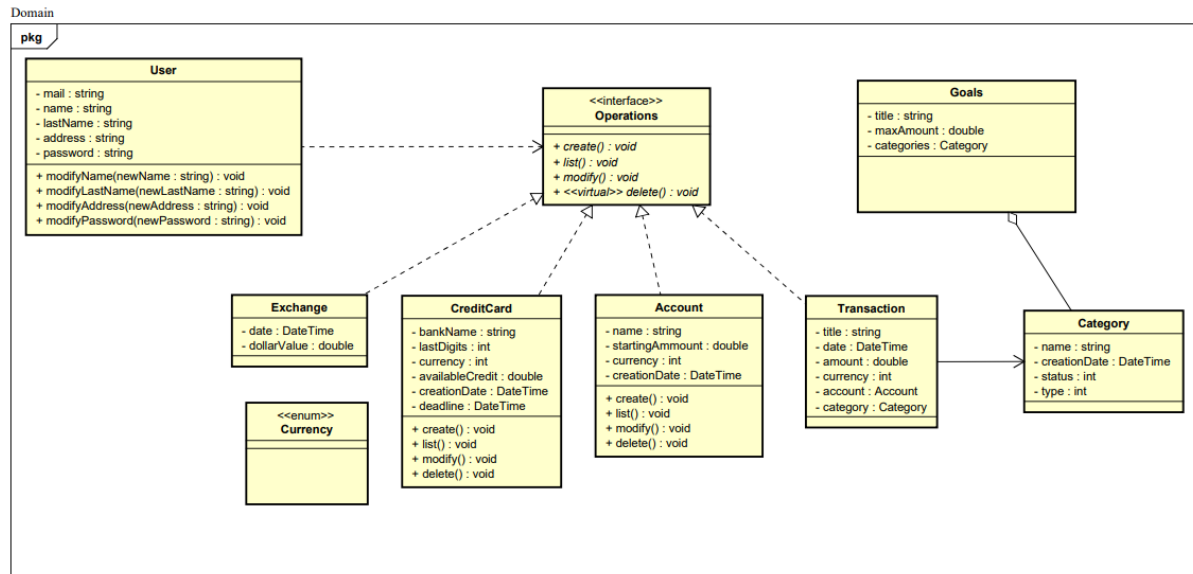
- Invitar usuario a un workspaces
- Crear workspace
- Generar reportes
- Ver listado de usuarios con acceso al workspace
- Agregar múltiples categorías a un objetivo
- Eliminar workspace
- Eliminar cuenta de usuario

Los bugs conocidos son:

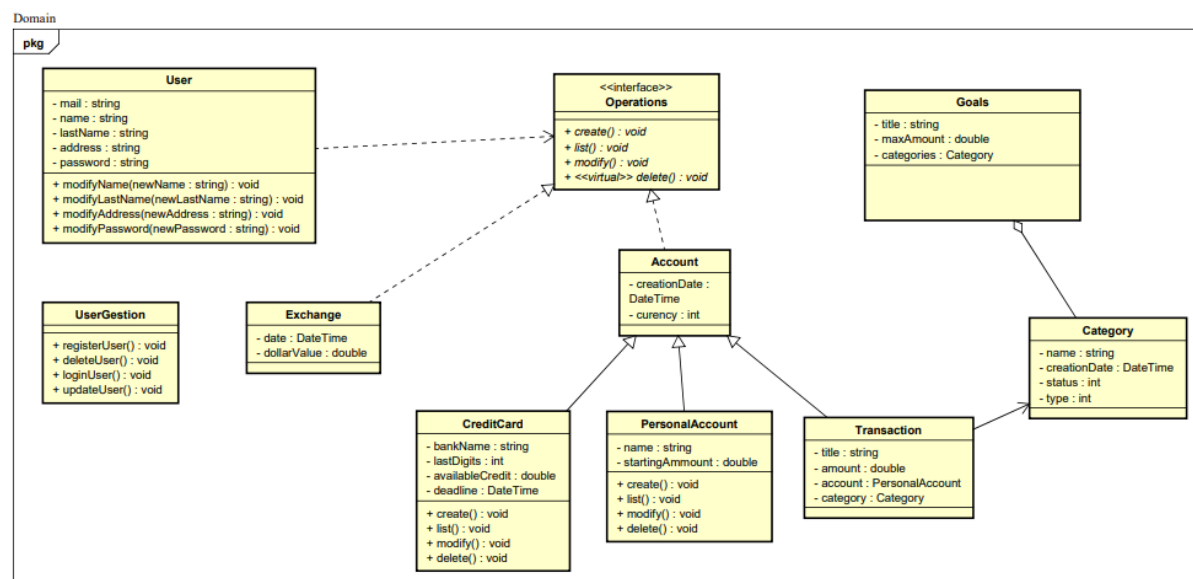
- Al editar los últimos 4 dígitos de la tarjeta y asignarle letras, el front end notifica que se cambió, pero el backend no lo hace porque no es correcto. Entonces pareciera que sí quedo editado.

DISEÑO Y DIAGRAMAS:

En el primer contacto con el proyecto desarrollamos el siguiente diagrama:



Como se puede apreciar nuestro primer enfoque fue como implementar alguna interfaz vista en el teórico. Más adelante nos metimos con las clases y la evolución del UML fue la siguiente:



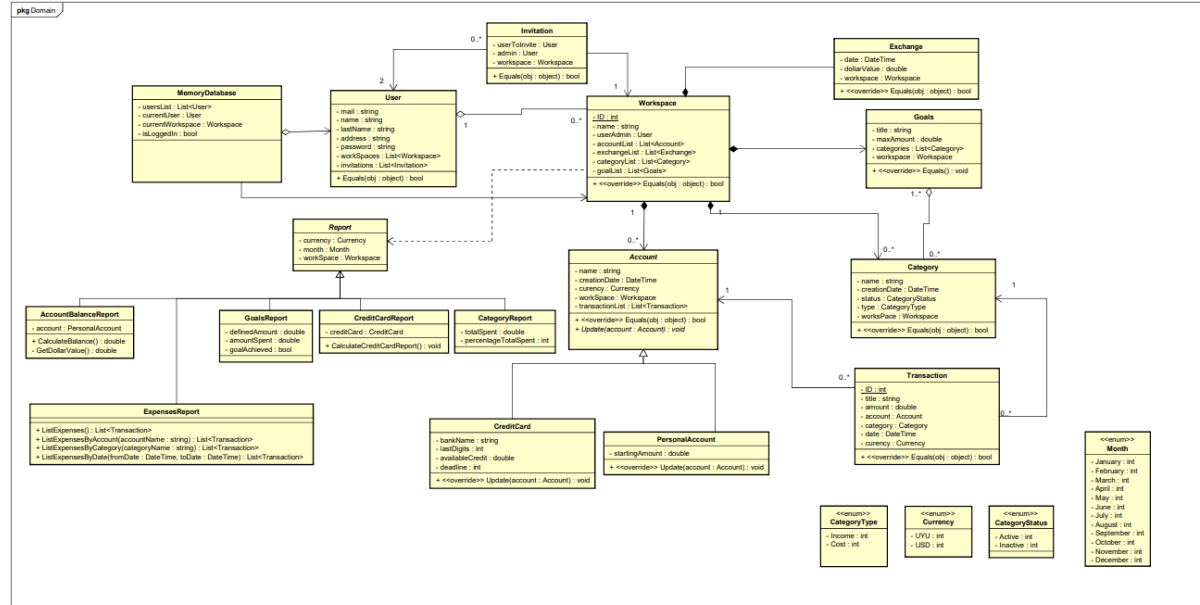
Aún nos faltaba diseñar cómo íbamos a hacer la gestión del usuario y el diseño de los reportes pero implementamos una herencia en cuentas para poder diferenciar entre los tipos.

Más adelante nos dimos cuenta de que las transacciones nos iban a representar un problema, entonces decidimos que no heredara de cuenta ya que no tenía mucho sentido.

Los diagramas finales son:

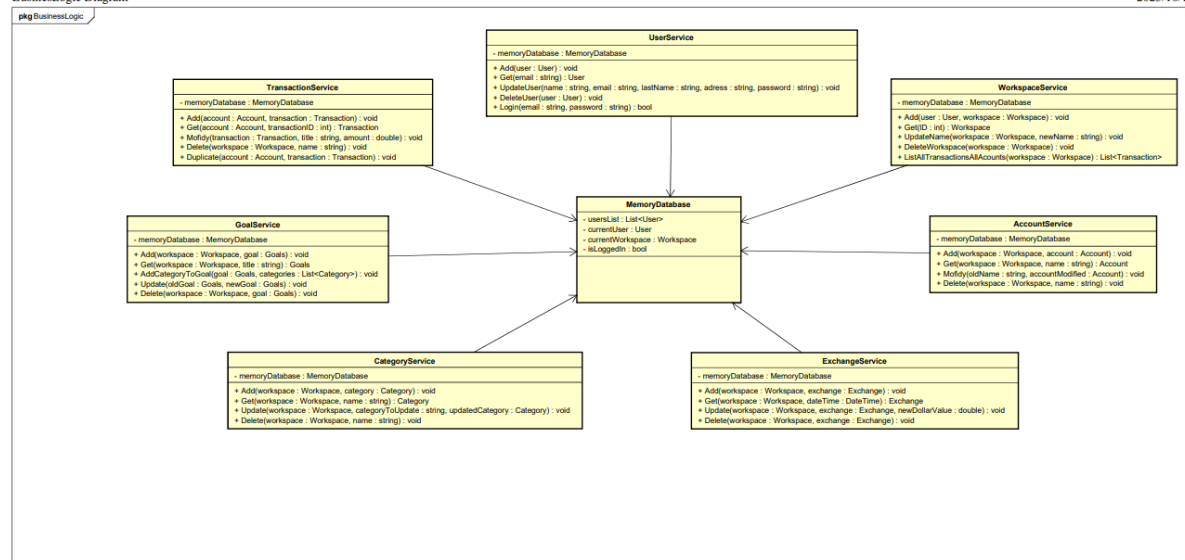
Domain Diagram

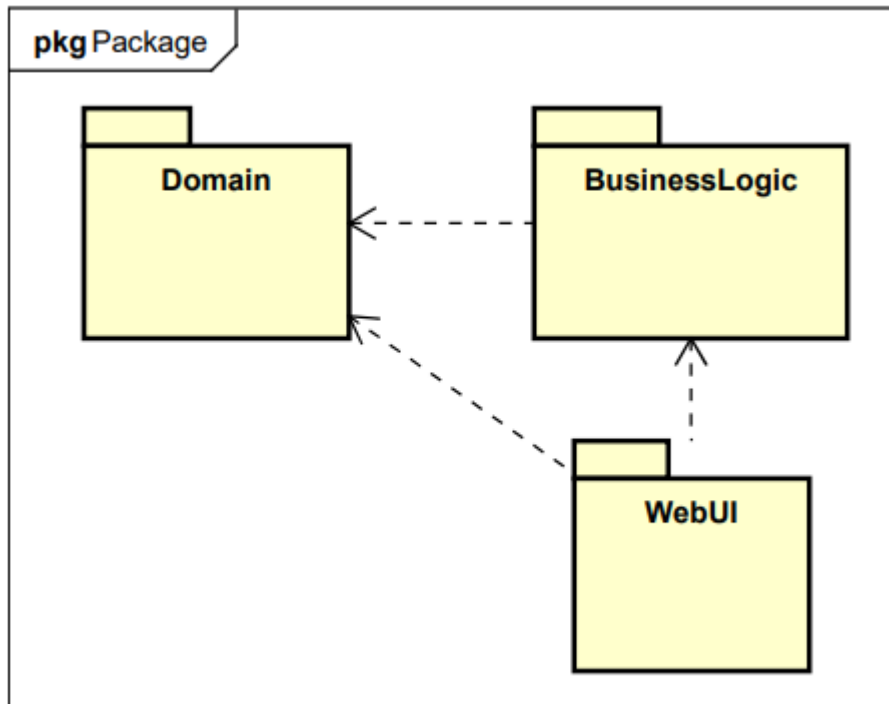
2023/10/12



BusinessLogic Diagram

2023/10/12





PRINCIPALES DECISIONES TOMADAS:

Decisiones tomadas para el diseño de la aplicación:

En la memoria solo se almacenará la lista de Usuarios, ya que a través de estos, se puede obtener toda la información, ya que los usuarios contienen los workspaces y dentro del workspace se almacena el resto de las listas.

Una de las decisiones más difíciles fue dónde guardar las listas. No sabíamos si guardar todas las listas por separado en la memoria, o que una lista contenga listas de las otras clases.

Decidimos guardar listas de algunas clases en otra clase, ya que para actualizar los datos era más sencillo y no generaba inconsistencia.

Otra de las decisiones que tomamos fue que si a un workspace, pueden acceder dos o más usuarios, el usuario administrador no podrá eliminar ese workspace. Al momento de eliminarlo, solamente pierde el acceso a ese workspace, y se asigna cualquier otro usuario con acceso como usuario administrador.

Las cuentas, transacciones, objetivos, categorías y reportes, pertenecen a un Workspace, y no se relacionan con el usuario. Esto se decidió ya que si un usuario desea eliminar su cuenta, solamente se borre el usuario de la memoria y toda la información siga estando en el workspace.

En caso de que al workspace solamente acceda ese usuario, al borrar la cuenta de usuario, o el workspace, se elimina toda la información.

Otra decisión tomada fue que para agregar una Transacción, al menos tiene que haber un tipo de cambio con una fecha anterior a la fecha de la transacción registrado en el workspace. Con que haya uno ya es suficiente, ya que si no se cargaron tipos de cambio más recientes, significa que el valor del dólar no varió desde el último ingresado.

De los tipos de cambio solo se pueden modificar el valor del dólar para ese día, ya que si modificamos la fecha, podría generar inconsistencia para generar los reportes.

Relacionado con el párrafo anterior, si hay una transacción con fecha posterior a un tipo de cambio, este no se podrá eliminar, ya que para generar los reportes generará inconsistencia, sin importar el tipo de moneda de la transacción.

Las transacciones las identificamos por un ID que se autoincrementa, por lo que a la hora de duplicar una transacción, no se genera ninguna inconsistencia, ya que la fecha de la transacción siempre será mayor o igual a la de la transacción a duplicar.

Algo similar consideramos para las categorías. De las categorías se pueden editar todos los atributos, pero si existe al menos una transacción que use determinada categoría, no se podrá editar ni el estado (Activa, Inactiva) , ni el tipo (Ingreso, Egreso).

Con respecto a las cuentas, sí se pueden eliminar, y cuando se elimina una de estas, también se eliminan todas las transacciones registradas para esa cuenta.

Para algunos atributos de determinadas clases se decidió usar Enums, permitiendo que el código sea extensible. Por ejemplo si se quiere registrar transacciones en otro tipo de moneda, o agregar un nuevo estado a las categorías, solamente habría que agregar un nuevo valor al enum, y hacer las pruebas correspondientes.

La interfaz de usuario interactúa con el dominio a través de un patrón Singleton. Los servicios que se inyectan tienen el atributo memory Database en donde se guardan todos los datos. Esto nos garantiza que hayan únicas instancias y que todas las clases interactúen allí.

Para realizar las clases de Servicio, se intentó usar una interfaz, pero por tema de tiempos y dificultad para implementarla, finalmente no se utilizó.

Vemos que los métodos de todos los servicios son similares, pudiendo haber creado una interface y que cada servicio la implemente.

COBERTURA Y FUNCIONALIDADES:

Detalle de la cobertura de pruebas unitarias con su debido análisis y justificaciones. En caso de que la cobertura de líneas de código no supere el 90% para alguna funcionalidad, se deberá incluir un análisis sobre el porqué, explicando que faltó en los test para que no se ejercitara la parte de código no cubierto

Hierarchy	Covered (%Lines)	Not Covered (%Lines)	Covered (%Blocks)	Not Covered (%Blocks)	Partially Covered (%Lines)
Seba_DESKTOP-DHOQU6U_2023-10-12.20_16_36.coverage	92,12%	7,67%	93,79%	6,21%	0,21%
testbusinesslogic.dll	96,75%	3,25%	96,95%	3,05%	0,00%
testdomain.dll	93,42%	6,44%	94,62%	5,38%	0,14%
domain.dll	94,28%	5,57%	95,58%	4,42%	0,15%
businesslogic.dll	78,94%	20,28%	87,01%	12,99%	0,79%

La cobertura de los test es de 92% de las líneas de código. A nuestro entender esto demuestra que se utilizó TDD para el desarrollo del sistema. Todos los casos de uso fueron pensados y discutidos con el equipo con el fin de abarcar todo lo posible.

Link de video de Youtube:

explicando las funcionalidades del sistema:

<https://youtu.be/lkk8yqU5t6M>