

Laboratorio 4, 5 y 6

Universidad nacional de Colombia – Ing. De Software

Manuel David Medrano Monroy

Jherson Adrian Medina Correa

Nicolas Eduardo Pardo Arias

2019 – 2

Grupo 1

I). Identificación de archivos modificados.

Model:

- User.java: se agregaron nuevos atributos junto sus respectivos getter y setters, esto con el fin de cumplir con las especificaciones propuestas en el laboratorio #2.

Repository:

- En esta capa no se realizó ningún tipo de modificación en los archivos preexistentes.

DAO:

- En esta capa no se realizó ningún tipo de modificación en los archivos preexistentes.

Presentation:

❖ Presentation.activity:

- No se realizaron cambios al archivo MainActivity.java.

❖ Layout:

- No se realizaron cambios al archivo Activity_main.xml.

II). Identificación de archivos agregados.

Model:

- Movement.java: esta es la clase que representa los movimientos dentro de nuestra aplicación, contiene los atributos y métodos necesarios para cumplir las especificaciones das por el laboratorio #2.

Repository:

- MovementRepository.java: Se crea con el fin de establecer un puente entre la capa de base de datos y los modelos establecidos, en este caso el modelo que representa los movimientos dentro de la aplicación Movement.java. Contiene funciones que nos permiten crear y obtener movimientos (ya sea de manera individual o en grupo), todo desde la base de datos que maneja la aplicación.

DAO:

- MovementDao.java: funciona como interfaz de los atributos de tabla y funciones establecidas en MovementRepository.java.

Presentation:

❖ Presentation.activity:

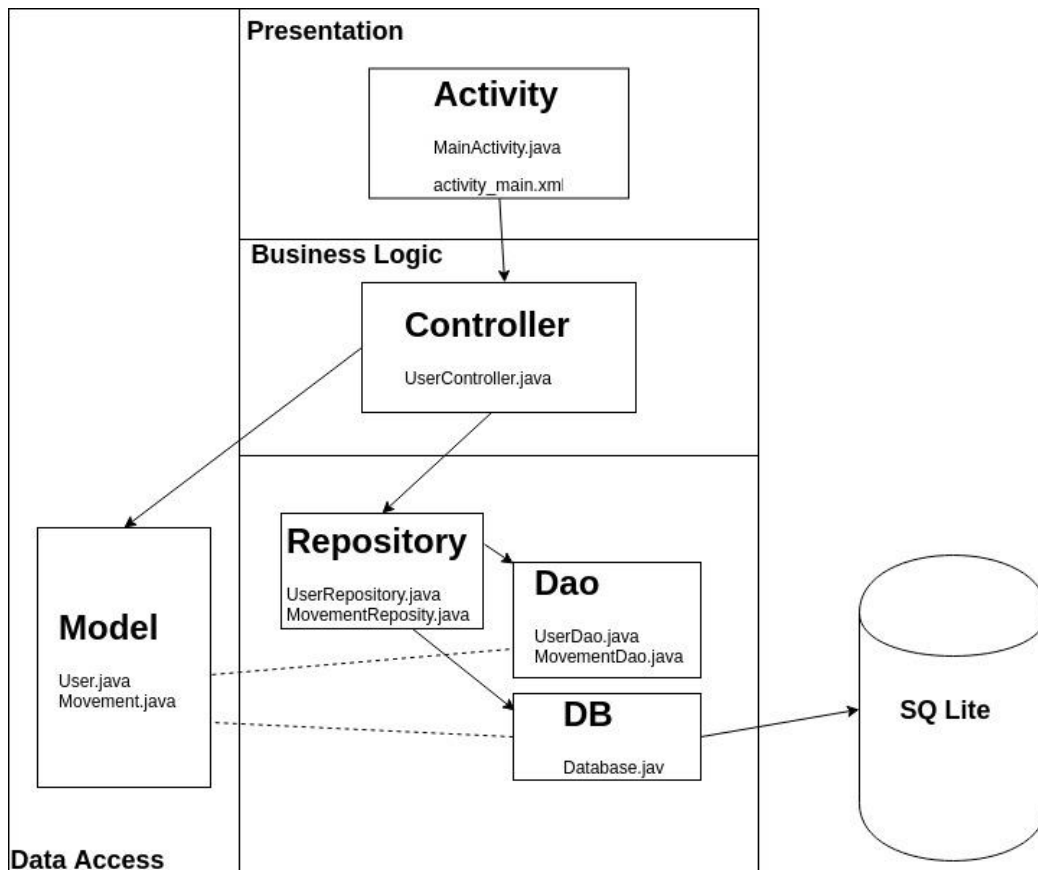
- No se agregaron archivos a esta capa.

❖ Layout:

- GUI1.xml: nuevo layout que contiene las características graficas presentadas al usuario dadas en el primer mockup de la interfaz gráfica del laboratorio #2.
- GUI2.xml: nuevo layout que contiene las características graficas presentadas al usuario dadas en el segundo mockup de la interfaz gráfica del laboratorio #2.

❖ III). Flujo de funcionalidad.

Con el fin de describir el flujo que se realiza dentro de la aplicación, haremos uso de uno de los diagramas facilitados por el profesor en clase, incluyendo en este los componentes que constituyen cada una de las capas:



Como vemos, las capas superiores como *Presentation* necesitan ayuda de capas inferiores para acceder a los datos y la lógica del negocio, así como la capa de *Business Logic*, para realizar las funciones necesarias para el funcionamiento y manejo de datos dentro de la aplicación debe hacer uso de controladores que se relacionen con los modelos y objetos establecidos en la capa de *Data Acces* los cuales se comunican con la información almacenada en forma de tablas en la base de datos.