MODEL VIEW PRESENTER (MVP)

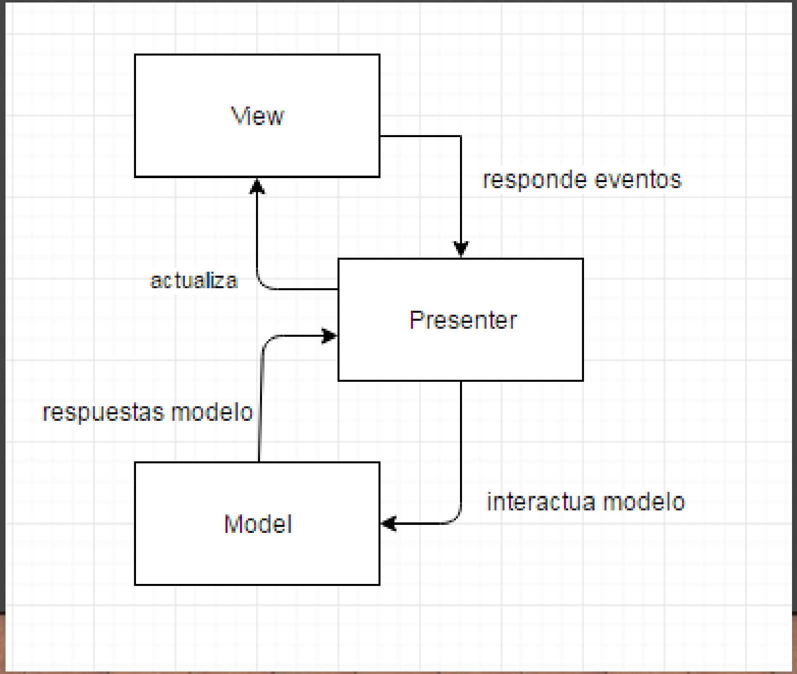
WORKSHOP EQUIPO R

NECESIDAD SEPARAR

- Acoplamiento entre datos y vista: CursorAdapters, ASyncTasks dentro de Activities o Fragments
- Claridad en el código, mantenibilidad
- Como parte de una arquitectura para las aplicaciones Android
- Responsabilidades

QUE ES MVP?

- Patrón para la capa de presentación
- Permite separar la capa de presentación de la lógica



IMPLEMENTACIÓN ANDROID

```
public interface LoginView {

    void showProgress();

    void hideProgress();

    void showMessage(String message);
}
```

```
public interface Presenter {

    void onResume();

    void onPause();

    void onDestroy();
}
```

IMPLEMENTACIÓN ANDROID

```
public class LoginPresenter {

   private LoginView loginView;

   public LoginPresenter(){}

   public void setView(@NonNull LoginView loginView) {
      this.loginView = loginView;
   }

   public void doLogin(String user, String pass){
      loginView.showProgress();
      performLogin(user, pass);
   }

   void onResume(){
      if (loginView != null) {
         loginView.showProgress();
      }
   }

   void onPause(){}
}
```

```
 private void performLogin(String user, String pass){
     AsyncTask<String, Void, String> doLoginTask = new AsyncTask<String, Void,
String>() {
        @Override
        protected void onPreExecute() {
           super.onPreExecute();
           loginView.showProgress();
        }
        @Override
        protected String doInBackground(String... params) {
           String user = params[0];
           String password = params[1];
           try {
              Thread.sleep(3000);
           } catch (InterruptedException e) {
              e.printStackTrace();
           }
           return "Login Ok!!!";
        }
        @Override
        protected void onPostExecute(String message) {
           super.onPostExecute(message);
           loginView.hideProgress();
           loginView.showMessage(message);
        }
     };
     doLoginTask.execute(user, pass);
  }
```

IMPLEMENTACIÓN ANDROID

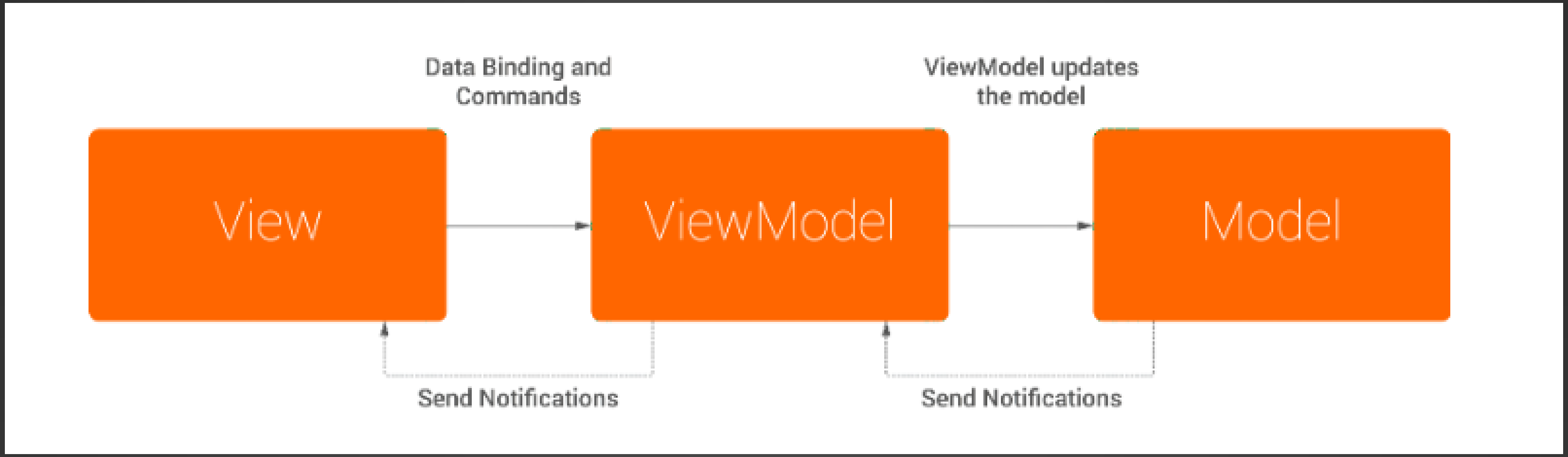
```
 public class LinearLayoutActivity extends AppCompatActivity implements LoginView{
   private LoginPresenter loginPresenter;
@Override
   public void showProgress() {
      if(progressBar!=null){
         progressBar.setVisibility(View.VISIBLE);
      }
   }

   @Override
   public void hideProgress() {
      if(progressBar!=null){
         progressBar.setVisibility(View.GONE);
      }
   }

   @Override
   public void showMessage(String message) {
      Toast toastMessage = Toast
            .makeText(this, message, Toast.LENGTH_LONG);

      toastMessage.show();
   }
}
```

MODEL VIEW VIEW MODEL (MVVM)



REFERENCIAS

- https://github.com/konmik/konmik.github.io/wiki/Introduction-to-Model-View-Presenter-on-Android
- https://github.com/pedrovgs/EffectiveAndroidUI
- http://antonioleiva.com/mvp-android/