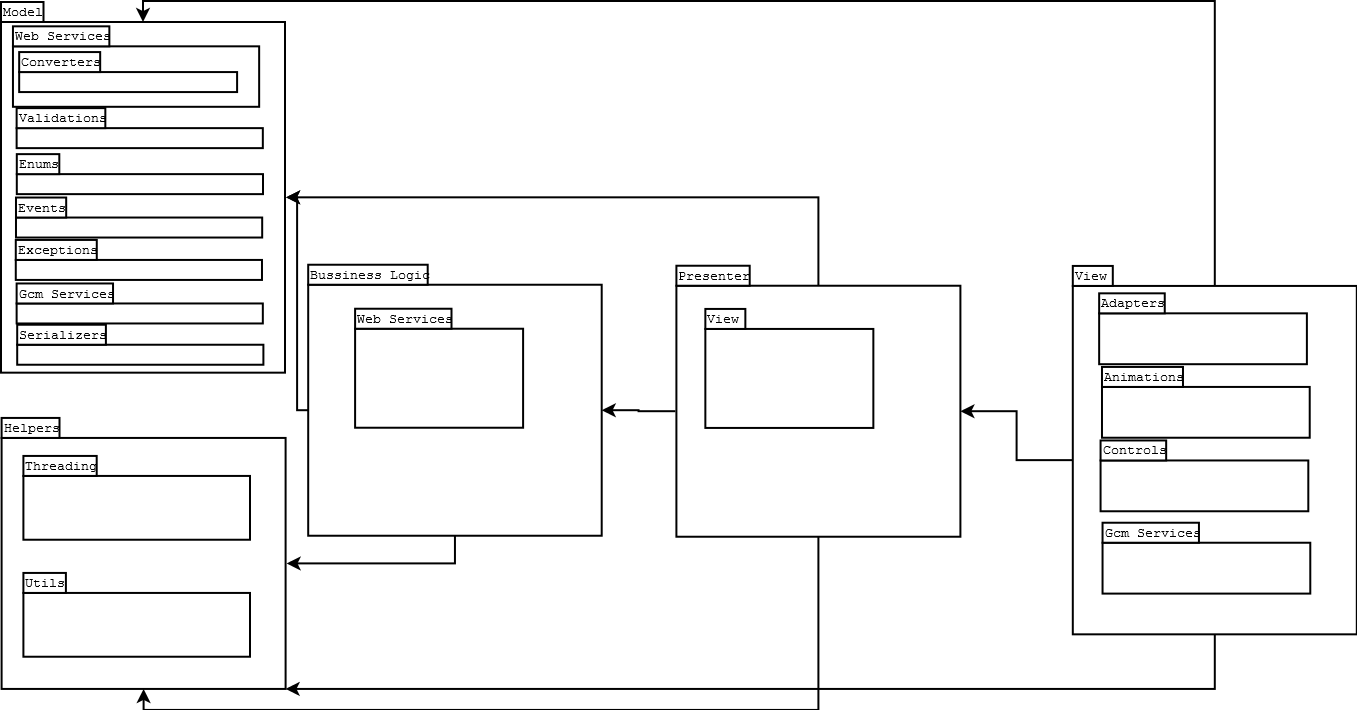
**Arquitectura del Sistema**

****

**View**

Capa de interacción directa con la interfaz de usuario, los elementos pertenecientes a esta deben limitarse a acomodar los datos para el usuario y no hacer nada de lógica de negocio.

Los miembros deben ser implementaciones de las interfaces definidas en la subcapa view (véase Presenter) cumpliendo estrictamente las tareas especificadas.

* **Adapters:** Son implementaciones personalizadas de adaptadores de listas o algún otro objeto de interfaz de Android, con el fin de tener una interfaz totalmente personalizada.
* **Animations:** Son implementaciones de animaciones como ser transiciones de pantalla, texto entre otros con el fin de hacer la interfaz más amigable y entretenida para el usuario final.
* **Controls:** Implementación de controles personalizados dándole forma, color, etc. Adecuados y convenientes para la aplicación.
* **GCM Services:** Clases para la gestión de notificaciones push dentro del dispositivo.

**Presenter**

Capa encargada de la preparación de datos para posteriormente ser desplegados por pantalla y también de recibir solicitudes desde la vista.

* **View:** Definición de interfaces que deberán ser implementadas por la vista.

**Bussines Logic**

Capa encargada de definir y hacer cumplir la lógica de negocio ya pre-establecida evitando inserciones dobles de datos y apoyando al momento de recuperarlos.

* **Web Services:** Sub capa encargada de la conexión con los servicios web que brindaran la mayor parte de la información utilizada en el sistema.

**Model**

Capa contenedora de las entidades identificadas además de definir la base de datos, podrá ser accedida desde cualquier parte del sistema con el fin de facilitar la abstracción.

* **Web Services:** Contenedora de clases necesarias para convocar y hacer uso de los datos devueltos por el servidor.
  + **Converters:** Clases encargadas de la conversión de datos en formato JSON devueltas por el servidor a clases propias del sistema.
* **Validations:** Contenedora de clases encargadas de implementar métodos de validación de campos como ser textFields.
* **Enums:** Contenedora de Enums para hacer el código mas entendible y mantenible.
* **Events:** Definición de eventos propios al ocurrir alguna acción que lo requiera en el sistema.
* **Exceptions:** Definición de excepciones propias.
* **GCM Services:** Clases de apoyo para la conexión con el servidor GCM
* **Serializers:** Implementacion de serializadores propios.

**Helpers**

Clases de apoyo al desarrollo del sistema, ya sea en formato, conversores genéricos entre otros.

* **Threading:** Clases de apoyo en la gestión de hilos.
* **Utils:** Utilidades en general.