**Capas de aplicación**

La arquitectura implementada es MVP donde se aprecian las siguientes capas principals:

1. **Capa Model:** responsable del manejo del modelo de negocio,
   1. **capa de persistencia:** almacenado en BD, consulta a BD
   2. **Capa de servicios:** es una fachada donde se construyen las consultas al api o a la Base de Datos en caso de no haber conexion
   3. **Capa DAO o capa de acceso al api “themoviedb:** consumir el api “themoviedb”
2. **Capa Views:** responsable de manejar todas las vistas de la aplicacion
3. **Capa Presenters:** responsable de la comunicación con el modelo para alimentar con datos las vistas.

**Clases**

**Capa de persistencia**

1. **DataController.swift: responsable de almanecenar y consultar a la base de datos**

**Capa de servicios**

1. **ServiceFactory.swift:** realiza la creacion de objetos abtractos de la capa de servicios
2. **MovieService.swift:** protocolo del servicio que se debe implementar
3. **TheMoviedbService.swift:** implementacion del sevicio donde se realizan todas las consultas al API o a la base de datos (listar categorias, consultar cada categoria y sus peliculas, consultar detalles de las peliculas, consultar imagines, buscar peliculas.

**Capa DAO**

1. **CategoryDaoFactory.swift:** creacion de objetos abstractos para el DAO de categories
2. **MovieDaoFactory.swift:** creacion de objetos abstractos para el DAO de movies
3. **CategoryDao.swift:** protocolo del DAO de categories que se debe implementar
4. **CategoryDaoImp.swift:** esta clase es responsible de consultar al api las categorias y sus peliculas
5. **MovieDao.swift:** protocolo del DAO de movies que se debe implementar
6. **MovieDaoImp.swift:** responsable de consultar los detalles de una pelicula, las imagines de una pelicula y de buscar pelicula

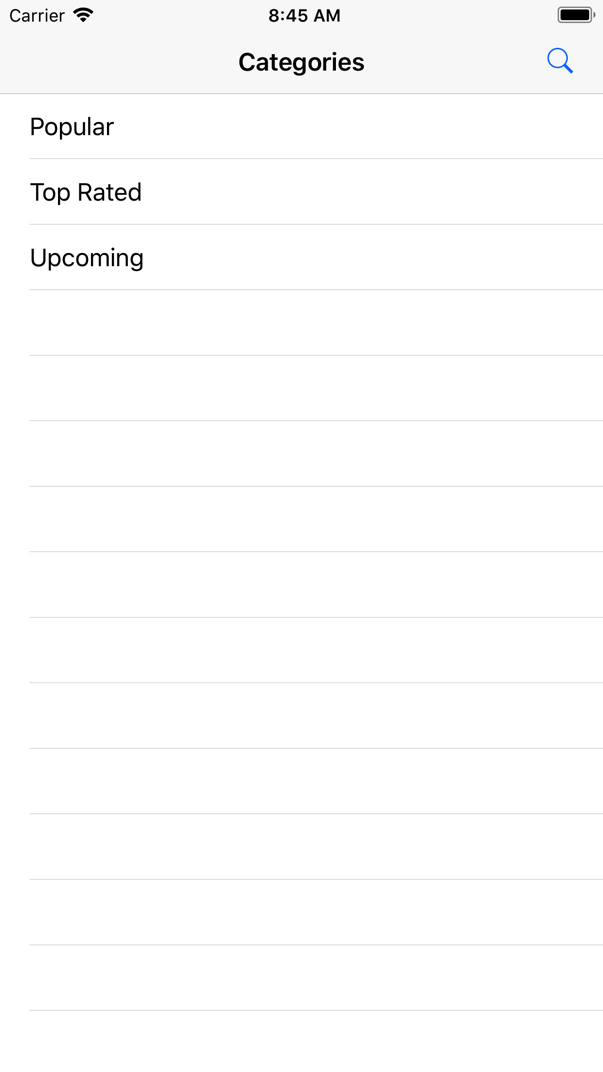
**Capa Views**

1. **CategoryTableViewController.swift:** es la vista encargada de mostrar las categorias
2. **MoviesViewController.swift:** esta vista es responsable de mostrar las peliculas de X categoria
3. **MovieCollectionViewCell.swift:** es un tipo de celda personalizado para dar estilo a la presentacion de una pelicula
4. **MovieDetailViewController.swift:** muestra mas detalles de la pelicula
5. **SearchMovieViewController.swift:** permite realizar y mostrar la busqueda de cualquier pelicula

**Capa Presenters**

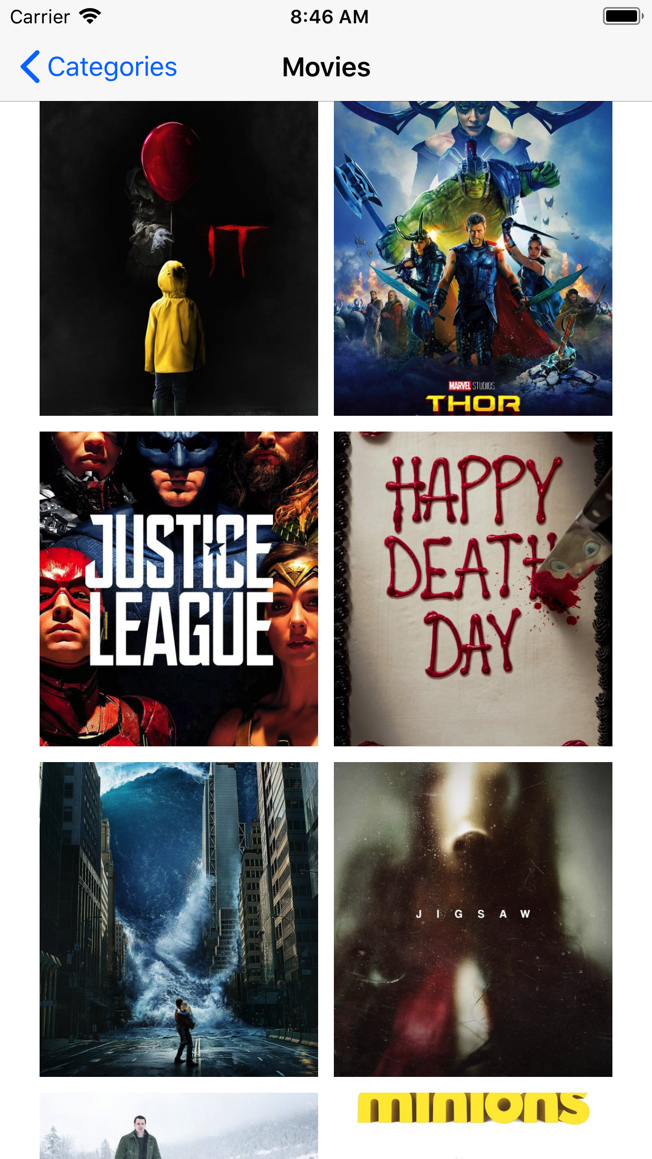
1. **CategoryPresenter.swift:** encargado de traer y enviar a la vista la lista de categorias
2. **MoviesPresenter.swift** encargado de traer y enviar a la vista la lista de peliculas
3. **MovieDetailPresenter.swift:** encargado de consultar los detalles de una pelicula para pasar los datos a la vista de detalle
4. **MovieSearchPresenter.swift:** este presentador realiza la consulta de peliculas y las envia a la vista para ser mostradas y permite desplegar el detalle

**DESCRIPCION DEL APP**

****

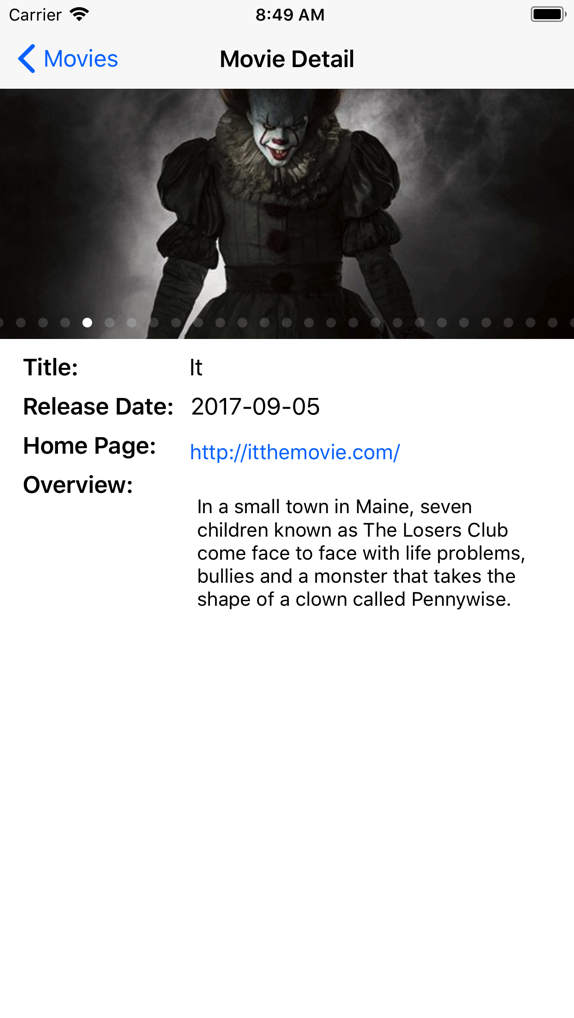
**vista de categorias:**

* **esta vista permite listar las 3 categorias requeridas**
* **el boton superior derecho de busqueda, permite abrir la vista de busqueda de peliculas**



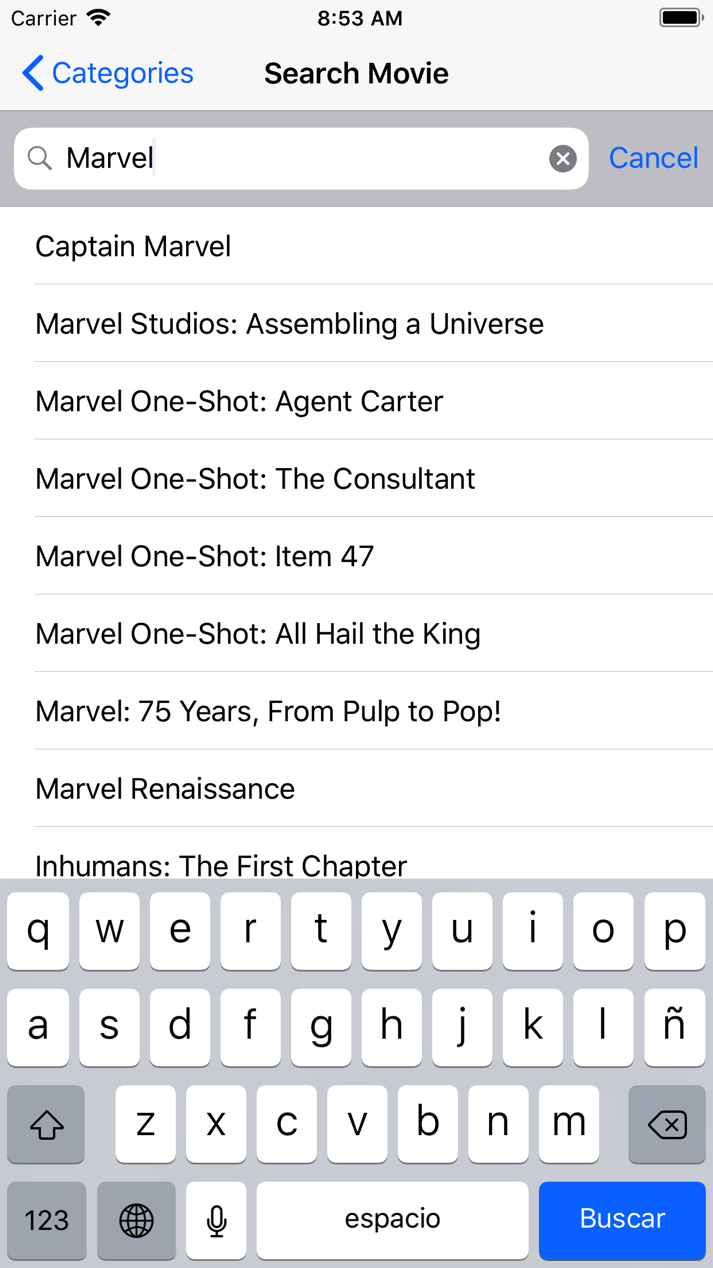
**vista de películas:**

* esta vista permite listar todas las películas de una categoría, al llegar a la ultima celda del collectionview se consultan mas películas ya que el servicio viene paginado.
* También al tocar una película el usuario puede ver el detalle



**Vista de detalle:**

* esta vista permite ver mas detalles de una película, además muestra todas las imágenes disponibles de la película con un scroll automatico, dichas imágenes se pueden ampliar. Al tocar los links o urls de home page se abre el navegador safari con el contenido de la pagina “Home Page”

****

**vista de búsqueda:**

* esta vista permite buscar películas, también cuenta con paginación y permite ver el detalle de las películas al tocar sobre una de ellas