

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

**FACULTAD DE CIENCIAS E INGENIERÍA
ESPECIALIDAD DE INGENIERÍA INFORMÁTICA**



FlashApp

Documento de Arquitectura de Software

Versión 1.0

ELABORADO POR:

20120383	Almendra Ximena Nuñez Calderón	almendra.nunezc@pucp.pe
20120081	Pedro Martín Vasquez Baldassari	martin.vasquezb@pucp.pe
20125934	Jean Piere Sullón Monteza	ipsullon@pucp.pe
20131220	Alexandra Huamaní Parede	a.huamani@pucp.pe
20135298	Gonzalo Ivan Campos Acosta	a20135298@pucp.pe

Lima, 5 de Mayo de 2018

Historial de Revisiones				
Ítem	Fecha	Versión	Descripción	Equipo
1.0	17/04/2018	1.0	Doc. Arquitectura de Software	Grupo 1

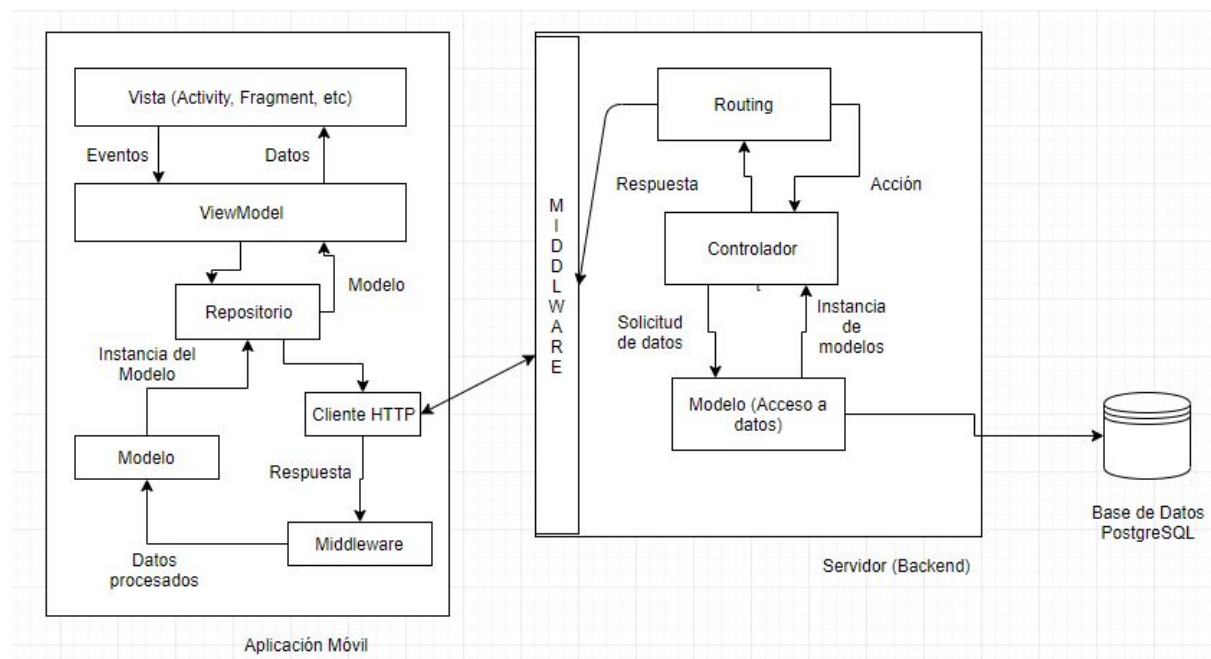
Tabla de Contenido

Objetivo	3
Representación de la arquitectura	3
Vista de Implementación	4
Performance	4

Objetivo

El objetivo del presente documento es mostrar la especificación de la arquitectura a seguir para la aplicación móvil FlashApp.

Representación de la arquitectura



Esta vista muestra el detalle de las capas en la arquitectura.

1. Arquitectura global

1.1. Capa de Acceso a datos: Servidor de Base de Datos

Capa en la que se gestiona el almacenamiento y acceso a los datos.

Está compuesta por un motor tipo SQL que se encarga de procesar las solicitudes de acceso, consulta o actualización desde la capa de negocio.

1.2. Capa de Negocio: Servidor de aplicaciones

Capa donde residen los controladores de la aplicación. Se utilizará un servidor Laravel y se comunicará con la aplicación por medio de servicios REST.

Recibe peticiones de la capa de presentación las cuales procesa y, de ser necesario, llega a consultar la capa de datos para resolver las peticiones.

Esta capa da soporte al Servidor de aplicaciones, las cuales manejan secciones distintas de la lógica de negocio.

1.3. Capa de presentación: Cliente móvil

Capa que sirve de comunicación entre la aplicación y el usuario real. Esta capa se comunica únicamente con la capa de negocio.

2. Arquitectura de la aplicación móvil

Para la implementación de la aplicación móvil, se ha elegido implementar el patrón de diseño MVVM (Model-View-ViewModel), en conjunto con el patrón de repositorio para el acceso a datos.

2.1. Vista

Recibe eventos de la interfaz y los envía al ViewModel.

2.2. ViewModel

Esta capa recibe los eventos generados por la interacción del usuario en la capa de Vista, y coordina los recursos necesarios para responder a la solicitud del usuario.

2.3. Modelo

Esta capa contiene una representación básica de los modelos de la base de datos.

2.4. Repositorio

Este componente de la arquitectura es el que obtiene los datos requeridos por el ViewModel a través de servicios, como el servidor de aplicación, o servicios externos.

2.5. Comunicación

Este componente permite la comunicación entre el backend y la aplicación por medio de JSONs

2.6. Middleware

El middleware se encarga de procesar las respuestas obtenidas a través del cliente HTTP, y transformarla en una instancia del Modelo, de manera que pueda ser utilizado por el ViewModel y la Vista.