Arquitectura

There is guidance within this template that appears in a style named InfoBlue. This style has a hidden font attribute that allows you to toggle whether it is visible or hidden in this template. Use the Microsoft® Word® menu **Tools > Options > View > Hidden Text** check box to toggle this setting. There is also an option for printing: **Tools > Options > Print**.

# Propósito

Este Documento de Arquitectura de Software (SAD) proporciona una descripción de la arquitectura del Sistema Gestor SAMOC; comprendiendo los procesos importantes: Gestionar Socios, Gestionar clientes, Gestionar Aportes, entre otros procesos de la Sociedad. Haciendo uso de diversas visiones arquitectónicas para representar diversos aspectos del Sistema.

Este documento tiene como objetivo, servir de referencia para el diseño y futura implementación, brindando a los implementadores y especialistas técnicos la información necesaria para el desarrollo del sistema.

# Objetivos de la Arquitectura y filosofía

* Brindar interoperabilidad entre los componentes de Software.
* Proporcionar Independencia y Flexibilidad entre la aplicación y los Servicios.
* Garantizar un alto nivel de escalabilidad, para el aumento de la capacidad de los componentes, en cualquier momento y balanceo de las cargas de los procesos.
* Facilitar el Mantenimiento, a través de la independencia de los cambios.
* Permitir la Reutilización de componentes y de aplicaciones.
* Utilizar eficientemente los recursos del hardware, ya que los componentes son distribuidos a través de toda la red.
* Minimizar la dependencia del Sistema de la plataforma en la que corre, lo cual garantiza la Portabilidad.

# Suposiciones y Dependecias

* Modelo arquitectónico MVC es una arquitectura orientada a la web para esta aplicación
* Se requiere que el cliente cuente con manejador de base de datos MySQL versión 5.0 o superior y servidor Web Apache 2, así como con los equipos de cómputo necesarios para el montaje del sistema de información.
* Se asume que los usuarios del sistema deben poseer conocimiento y habilidades en el ámbito de sus funciones: conocimiento de los procedimientos definidos por la institución.
* El sistema de información debe funcionar autónomamente, sin necesidad de comunicarse con otros sistemas externos a la institución, por lo que no hay dependencias respecto de otros sistemas y recursos.
* El sistema puede ser reparado en un tiempo razonable.

# Requisitos Arquitectónicamente significativos

* El sistema debe controlar los estados de cada consulta de un usuario
* El sistema debe responder a eventos automáticamente.
* El sistema debe parametrizar datos del modelo a la vista
* El sistema debe permitir resolver problemas independientemente de cada componente

# Las decisiones, limitaciones, y justificaciones

* Restricciones de Tiempo.
* Restricciones de Seguridad y Privacidad.
* Restricciones de Tecnología.

|  |  |
| --- | --- |
| **Limitación Arquitectónica** | Tiempo |
| **Código y Nombre** | 01-Restricción de tiempo |
| **Declaración del Problema** | Existe una fecha de entrega a la cual el proyecto debe estar terminada |
| **Supuestos** | Terminar a tiempo para la entrega final |
| **Motivación** | Aprender nuevas herramientas para el desarrollo de software |
| **Alternativas** | Aprovechar el tiempo para la ejecución del proyecto |
| **Decisión** | Realizar los entregables en el tiempo establecido |
| **Estatus** | En proceso |
| **Justificación** | El manejo de mucha información en poco tiempo genera demoras |
| **Implicaciones** | Entregables a tiempo. |

|  |  |
| --- | --- |
| **Limitación Arquitectónica** | Seguridad y Privacidad |
| **Código y Nombre** | 02-Restricciones de Seguridad y Privacidad |
| **Declaración del Problema** | Prestar toda la seguridad a los usuarios. |
| **Supuestos** | Poder acceder al sistema. |
| **Motivación** | Aprender nuevas herramientas para el desarrollo de software |
| **Alternativas** | Aprovechar de la fortaleza de la seguridad para los datos del usuario |
| **Decisión** | Realizar los entregables en el tiempo establecido |
| **Estatus** | En proceso |
| **Justificación** | La protección de la seguridad de la información de los datos |
| **Implicaciones** | Entregables a tiempo. |

|  |  |
| --- | --- |
| **Limitación Arquitectónica** | Tecnología |
| **Código y Nombre** | 03-Restricción de tecnología |
| **Declaración del Problema** | Usar la tecnología que está a nuestro alcance. |
| **Supuestos** | A través de la tecnología mejorar los procesos |
| **Motivación** | Aprender nuevas herramientas para el desarrollo de software |
| **Alternativas** | Aprovechar la tecnología para la ejecución del proyecto |
| **Decisión** | Realizar los entregables en el tiempo establecido |
| **Estatus** | En proceso |
| **Justificación** | Mejorar el proceso con la tecnología que tenemos |
| **Implicaciones** | Entregables a tiempo. |

# Mecanismos de la Arquitectura

## La arquitectura está representada siguiendo las recomendaciones de OPENUP, Las vistas necesarias se enumeran a continuación.

|  |  |  |
| --- | --- | --- |
| **VISTA** | **ELEMENTO DE MODELADO** | **DESCRIPCIÓN** |
| Caso de Uso | Diagrama | Describe el proceso de negocio más significativo y el modelo del dominio. Presenta los actores y los casos de uso para el sistema. |
| Vista de Restricciones |  | Describe restricciones tecnológicas, normativas, uso de estándares, entre otros, las cuales deben ser respetadas tanto por el proceso de desarrollo como por el producto desarrollado |
| Lógica | Diagrama de Clases | Describe la arquitectura del sistema presentando varios niveles de refinamiento. Indica los módulos lógicos principales, sus responsabilidades y dependencias. |
| Vista de Procesos |  | Describe los procesos concurrentes del sistema |
| Implementación | Diagrama | También llamada vista de desarrollo, describe la organización estática del software. |
| Modelo de Datos |  | Presenta los modelos de datos, los servicios de persistencia. |
| Implantación | Diagrama | También conocida como vista física y se encarga de modelar la disposición física del software. |

# Abstracciones Clave

* **Arquitectura de Software:** Vista abstracta de alto nivel del sistema, incluyen lo componentes principales con conducta que interactúan para alcanzar la misión del sistema, es un artefacto primario, conceptualizar, gerencia y construir la evaluación del sistema en desarrollo.
* **Modelo vista controlar (MVC):**

Se forma en un modelo, varias vistas, varios controladores entonces las vistas y los controladores están muy relacionados, los controladores tratan los eventos que se producen en la interface gráfica.

* **Laravel 5:** es un [framework](http://es.wikipedia.org/wiki/Framework) de [código abierto](http://es.wikipedia.org/wiki/C%C3%B3digo_abierto) para desarrollar aplicaciones y servicios web con [PHP](http://es.wikipedia.org/wiki/PHP) 5. Su filosofía es desarrollar código PHP de forma elegante y simple.
* **Vista de restricciones:**

Describe restricciones tecnológicas, normativas, uso de estándares, entre otros, las cuales deben ser respetadas tanto por el proceso de desarrollo como por el producto desarrollado.

* **Clase:**

Describe la arquitectura del sistema presentando varios niveles de refinamiento. Indica los módulos lógicos principales, sus responsabilidades y dependencias.

* **Vista de procesos:**

Describe los procesos concurrentes del sistema

* **Implementación:**

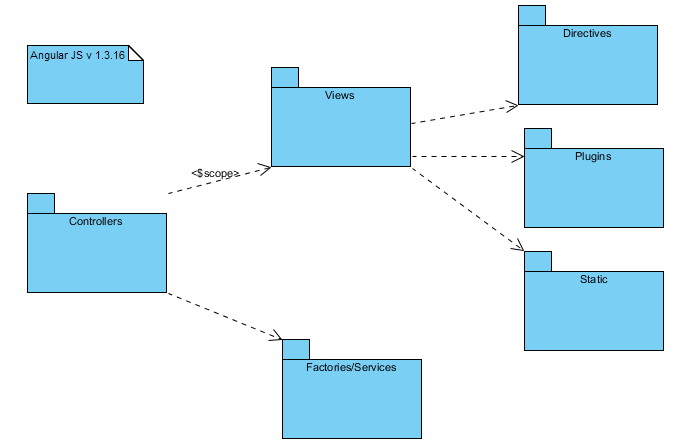
Actividad donde las especificaciones son llevadas a condigo de computadora.

* **Modelo de datos:**

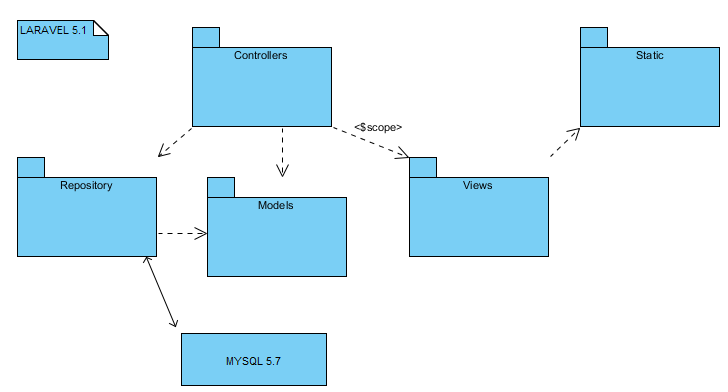
Presenta los modelos de datos, los servicios de persistencia.

# Capas o marco arquitectónico

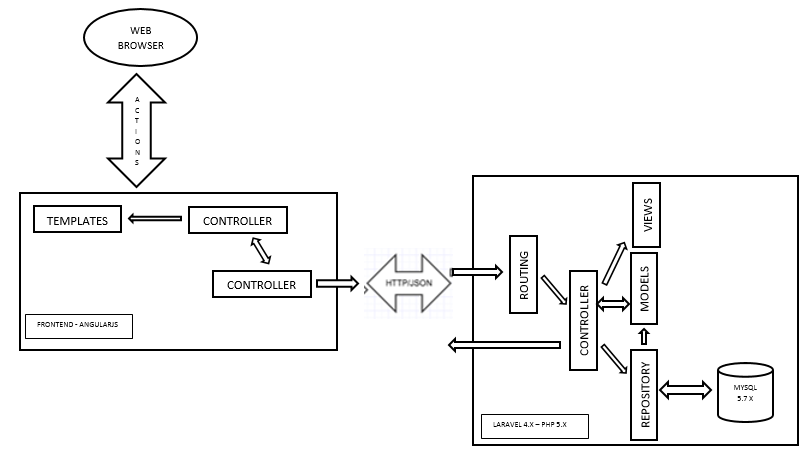
**8.1 FrontEnd**

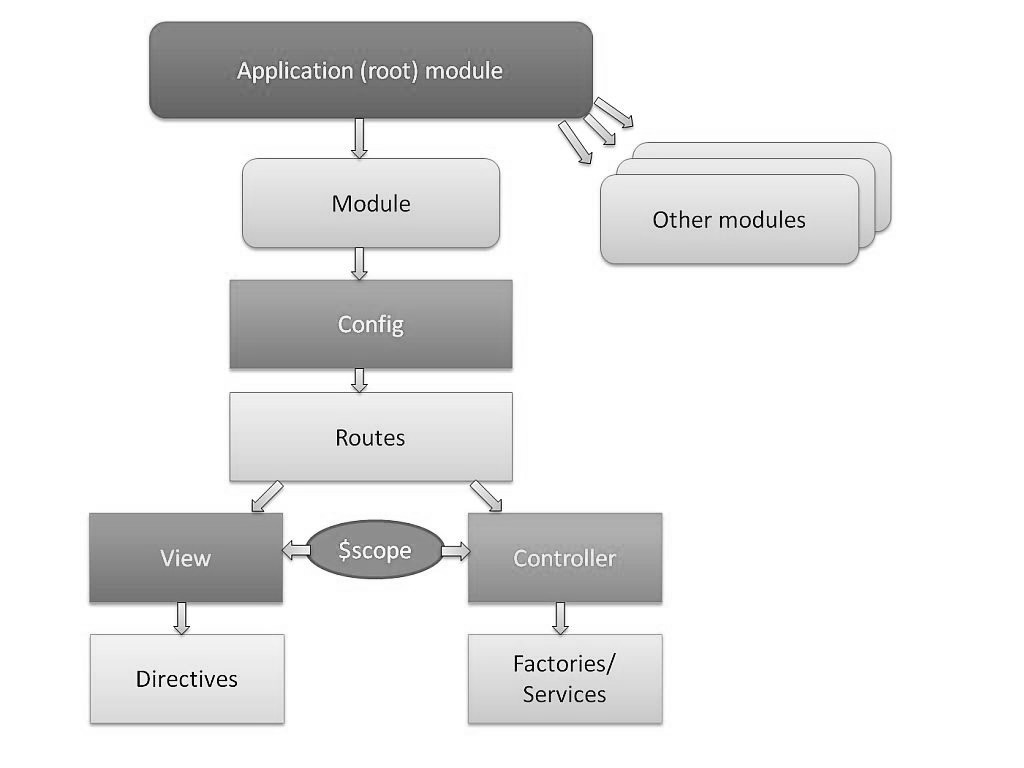


**8.2 BackEnd**



**COMUNICACIÓN FRONT-END – BACKEND (API)**





# Vistas de Arquitectura

A continuación se muestran las vistas en total en la representación arquitectónica del sistema.

## Vistas Recomendadas:

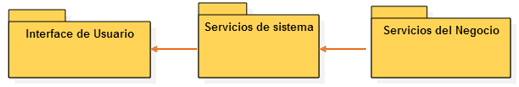
* **Vista Lógica**

Esta vista presenta tres niveles de Arquitectura; a continuación se grafican los niveles a tratar.



**Arquitectura del sistema:**

En el primer nivel se especifica el estilo de arquitectura para el sistema de gestión de información. El siguiente diagrama presenta la arquitectura del sistema.



Cada capa determina un rol para los módulos que residen en ella. La Interfaz de Usuario tiene como objetivo el manejo de la lógica del usuario. Está conformada por el conjunto de páginas web dinámicas que interactúan entre si y por módulos que rodean la lógica de los casos de uso.

Los Servicios del Sistema representan los servicios básicos que debe proveer el sistema; estos servicios son directamente utilizados por los módulos de la capa superior.

Los Servicios de Negocio son servicios de manejo de información del negocio; son servicios aún más básicos.

**Arquitectura Lógica:**

La Arquitectura Lógica presenta un refinamiento de la Arquitectura del Sistema. La dimensión Requerimientos, principalmente la Vista de Casos de uso, va a verse realizada por los módulos aquí presentados. Se analizará los módulos presentes en cada capa de la Arquitectura del Sistema, presentando finalmente la Arquitectura Lógica.

**Interface de usuario:**

Esta capa consiste de un módulo por cada caso de uso identificado. Cada módulo contiene la lógica que lleva adelante el caso de uso y un conjunto de páginas dinámicas utilizadas por dicha lógica.

**Arquitectura de Módulos:**

La Arquitectura de los Módulos presenta un refinamiento de la Arquitectura Lógica. Esta incluye, para cada módulo, una vista que mejor define su diseño. Para cada tipo de módulo, para los módulos de cada capa, se utilizará una vista diferente.

* **Vista de Casos de Uso:**

Esta vista presenta la percepción que tiene el usuario de las funcionalidades del sistema. Se presenta el proceso de negocio más importante del sistema y los casos de uso que se derivan de éste.

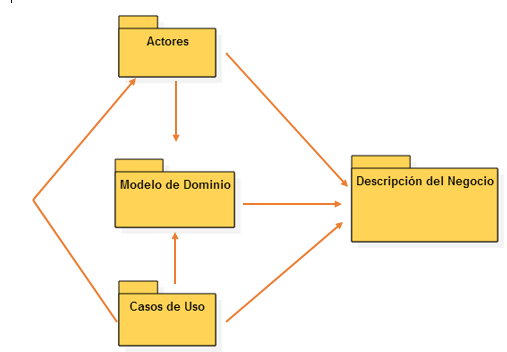


Diagrama Vista General

**DESCRIPCION DEL NEGOCIO:**

A continuación se describe el Sistema Gestor SAMOC, procesos del negocio y subprocesos que se relaciona.

**SISTEMA GESTOR SAMOC:** La Institución **“Sociedad Auxilio Mutuo Obrero – Cajamarca”** es una organización de derecho privado con personería jurídica, autónoma, apolítica, y sin fines de lucro, que promueve el desarrollo humano, cuyos miembros se rigen por el presente estatuto y su reglamento interno; cuya sigla es “SAMOC”.

El Sistema Gestor SAMOC será capaz de ayudar en las labores que se desarrollarán en relación a la Gestión de Socios y al alquiler de ambientes; este sistema contará con las funciones necesarias e importantes que requiere la Institución.

Maneja el tema de gestión de socios, se verá el registro del socio, el aporte mensual que realiza como también la entrega de beneficios mutuales a su heredero.

Sistema Gestor SAMOC trabaja con PHP y el framework Laravel 5, ambos crean el proyecto que se está desarrollando y logrando las funcionalidades de la Gestión de Socios y del Alquiler de ambientes.