Sistema de Automatización de registros en línea

Documento de la Arquitectura del Software

Versión 0.1

Historia de Revisión

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 19/05/2010 | 1.0 | Versión inicial del documento, previa a la Fase de Construcción del proyecto | |  | | --- | |  | |
| 01/06/2010 | 1.1 | Conversión de *Diagramas de Configuración Estándar del Sistema SAM, Componentes de Capa del patrón MVC* a UML. Actualizado *Diagrama de CU*. Agregado *Diagrama de Modelo de Dominio*. |  |
| 18/06/2010 | 1.2 | Actualización de *Diagrama de Modelo de Dominio y Diagrama de Casos de Uso*. Inclusión de *Diagrama ER, Diccionario de Datos*, *Diagrama de Clases, Diagramas de Secuencia.* |  |
| 14/07/2010 | 1.3 | Inclusión de los *Diagramas de Componentes* para las Capas *Modelo, Vista* y *Controlador.* Actualización de *Diccionario de Datos, Diagrama ER, Modelo de Dominio.* |  |

Tabla de Contenidos

1. Introducción 5

1.1 Propósito 5

1.2 Alcance 5

1.3 Definiciones, Siglas, y Abreviaciones 5

1.4 Referencias 5

1.5 Vista Global 5

2. Representación Arquitectónica 5

3. Metas y Restricciones Arquitectónicas 6

4. Vista de Casos de Uso 7

5. Vista Lógica 8

5.1 Visión general 8

5.1.1 Diagrama de Conceptual (Modelo de Dominio) 8

5.1.2 Diagrama de Clases 9

5.2 Realizaciones de los casos de uso 10

5.2.1 Crear Cuenta 10

5.2.2 Iniciar Sesión 10

5.2.3 Cerrar Sesión 11

5.2.4 Consultar Usuario 11

5.2.5 Modificar Usuario 12

5.2.6 Crear Proyecto 12

6. Vista de Implantación 12

6.1 Configuración Estándar 12

7. Vista de Implementación 13

7.1 Vista General 13

7.2 Capas 14

7.2.1 Vista 14

7.2.2 Controlador 15

7.2.3 Modelo 16

8. Vista de Datos 17

8.1 Diagrama Entidad-Relación (ER) 17

8.2 Diccionario de Datos 17

9. Tamaño y Desempeño 19

10. Calidad 20

Documento de la Arquitectura del Software

# Introducción

## Propósito

El propósito del documento presente tiene como propósito presentarles a los participantes del proceso de desarrollo del Sistema SAM una visión clara de los distintos aspectos que conforman la arquitectura del sistema. El Documento de Arquitectura del Software busca el aumento de la productividad del equipo desarrollador y sentar las bases de la arquitectura del software, con el fin de facilitar las actividades de análisis y toma de decisiones que puedan llevarse a cabo en un futuro.

## Alcance

Definir y unificar los criterios de desarrollo del Sistema SAM con respecto a los aspectos arquitectónicos del proyecto, mostrando los detalles sobre los paquetes, capas y procesos que conformaran el sistema.

## Definiciones, Siglas, y Abreviaciones

A continuación presentamos las abreviaturas y definiciones de los términos de mayor importancia que se encuentran en el documento.

**LISI:** Laboratorio de Investigación de Sistemas de Información.

**MECAP**: Método para Especificar CAsos de Prueba de software.

**MVC:** Modelo – Vista – Controlador. Estilo de arquitectura de software, frecuentemente visto en aplicaciones Web, que divide los procesos en tres capas.

**MySQL**: sistema de gestión de bases de datos relacional, distribuido como software libre.

**PHP**: lenguaje de programación interpretado, diseñado para la construcción de páginas web dinámicas.

**SAM**: Sistema Automatizado de MECAP.

**USB**: Universidad Simón Bolívar.

## Referencias

* Documento de Visión del Sistema (v 1.0 – 02/05/2010).
* Documento de Especificación de Requerimientos del Sistema (v 1.0 – 19/05/2010).

## Vista Global

En las siguientes diez (10) secciones del documento se detalla la arquitectura del Sistema SAM a desarrollarse. Se hará hincapié en los detalles de las distintas vistas (lógica, casos de uso, procesos, implementación e implantación) que conformaran la arquitectura del software (además de otros aspectos relevantes), empleando el lenguaje más sencillo posible y haciendo uso de diagramas donde se considere necesario con el fin de expresar las ideas de la forma más concisa.

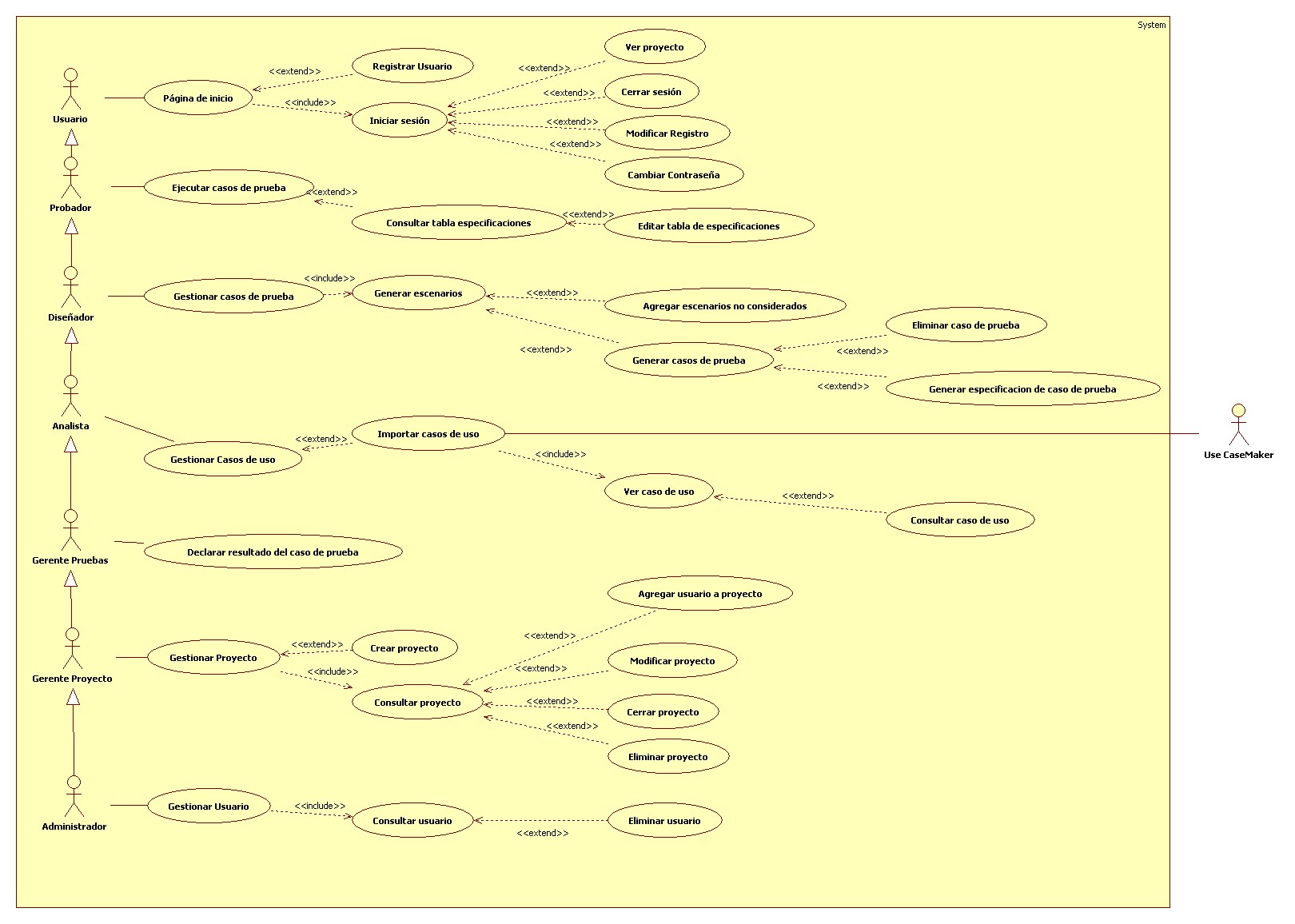
# Representación Arquitectónica

La representación arquitectónica del Sistema SAM se encuentra basada en el modelo de **4+1 Vistas** de Philippe Kruchten. En el transcurso del documento se tratara más a fondo los detalles de cada una.

# Metas y Restricciones Arquitectónicas

Con el desarrollo del sistema SAM se busca obtener una aplicación con un alto nivel de portabilidad, de fácil distribución y con un alto nivel de reusabilidad. En sus etapas iniciales, la aplicación será instalada en un servidor dentro de la USB, pero se busca obtener una pieza de software que se pueda instalar en cualquier servidor, con un mínimo de esfuerzo y con el mejor performance posible. Entre las restricciones principales del desarrollo del sistema SAM se encuentran: hacer uso de tecnologías de software libre (en nuestro caso, el lenguaje PHP y el manejador de base de datos MySQL) y el uso del patrón MVC para el desarrollo de la herramienta (impuesto por el LISI). Por último, se tiene como meta lograr que el sistema SAM pueda ejecutarse sin percances en los navegadores web Microsoft Internet Explorer y Mozilla Firefox, con el fin de poder serle útil a la mayor cantidad de usuarios posible.

# Vista de Casos de Uso

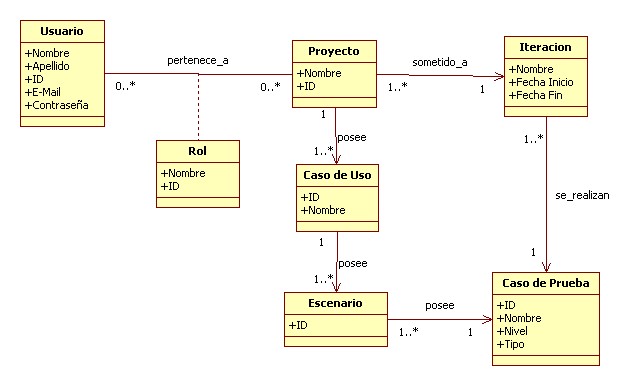


# Vista Lógica

## Visión general

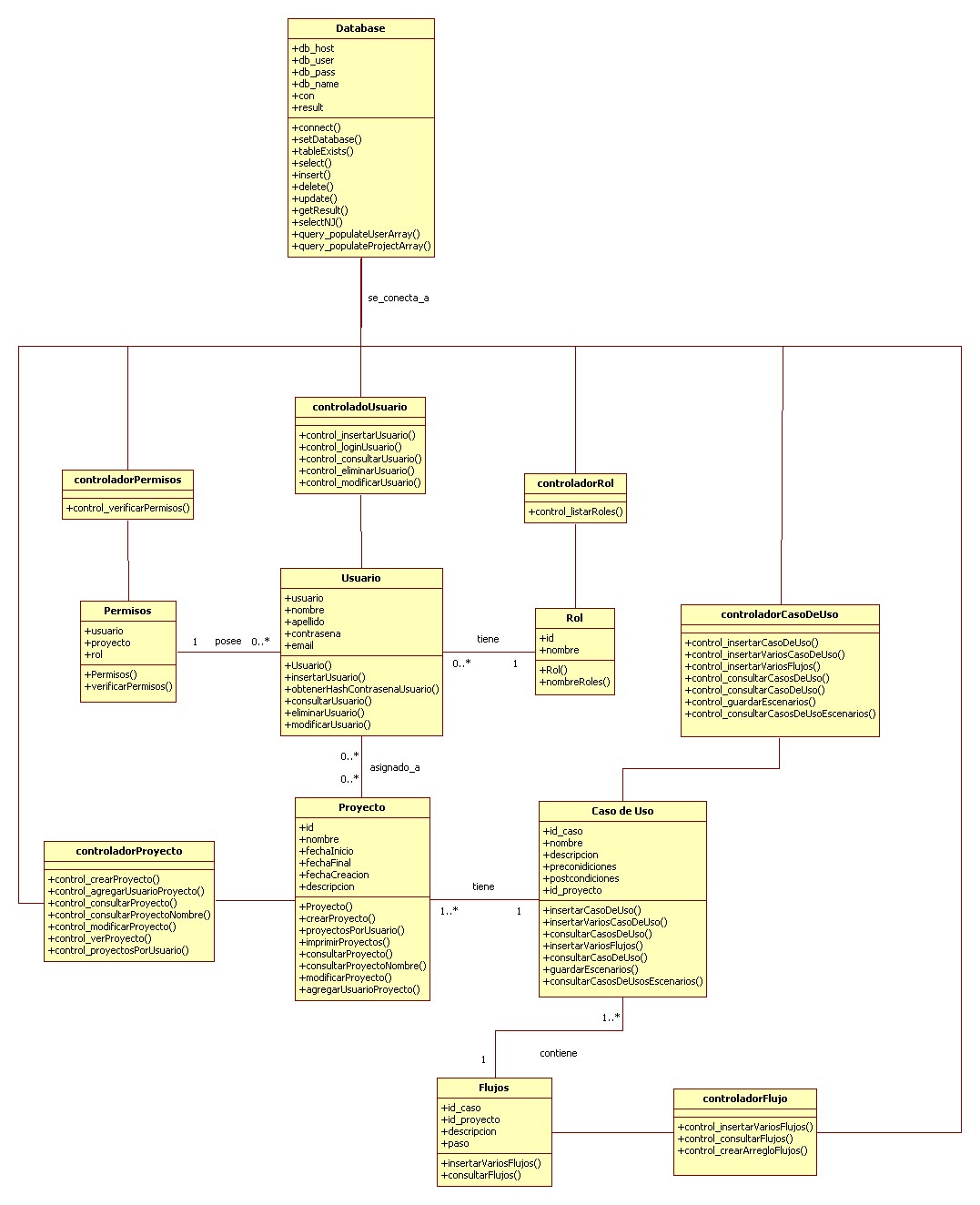
### **Diagrama de Conceptual (Modelo de Dominio)**

A continuación proponemos un Modelo Conceptual (diagrama de Modelo de Dominio) para demostrar cuales son los conceptos más relevantes y sus asociaciones para el Sistema SAM.



*Diagrama de Modelo del Dominio*

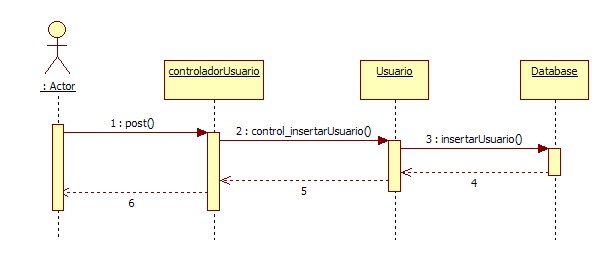
### **Diagrama de Clases**



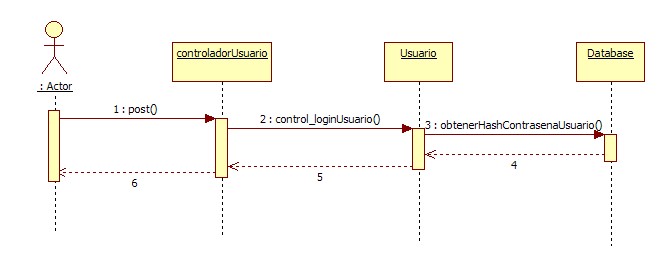
*Diagrama de Clases*

## Realizaciones de los casos de uso

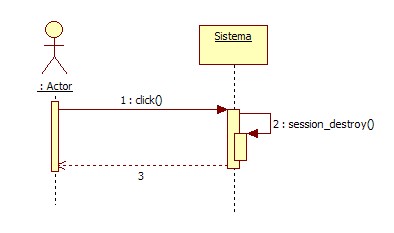
### **Crear Cuenta**



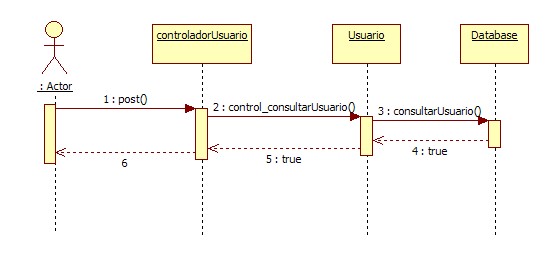
### **Iniciar Sesión**



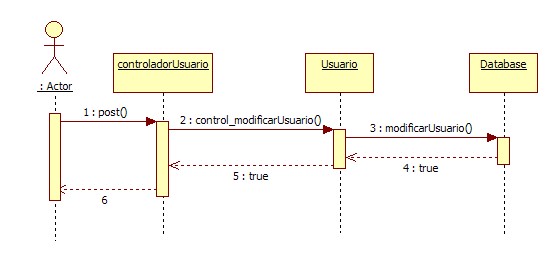
### **Cerrar Sesión**



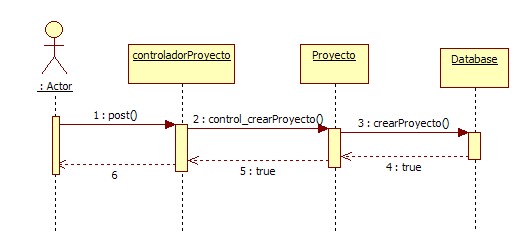
### **Consultar Usuario**



### **Modificar Usuario**



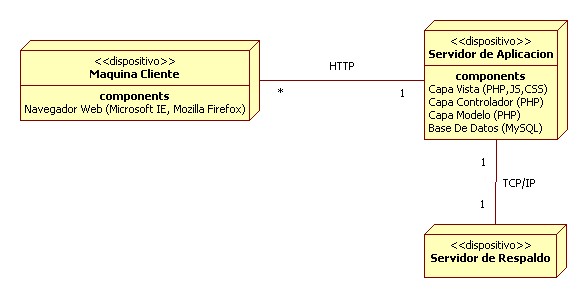
### **Crear Proyecto**



# Vista de Implantación

## Configuración Estándar

El Sistema SAM se está desarrollando con una configuración estándar sencilla. El Sistema SAM se hospedará y ejecutará desde un servidor principal, junto con el manejador de base de datos y los datos de la aplicación. Los usuarios tendrán acceso al sistema mediante el uso de estaciones de trabajo conectadas directamente al servidor (mediante una red interna, por ejemplo) o a través de Internet (asumiendo que conozcan la dirección I.P. del servidor en donde se ejecute el sistema SAM). Dichos clientes deben ejecutar un navegador web para poder hacer uso del sistema. Se recomienda tener un servidor de respaldo, el cual monitoree el servidor principal y pueda detectar cuando haya fallas en el mismo.



*Diagrama de la Implantación Estándar del Sistema SAM*

# Vista de Implementación

## Vista General

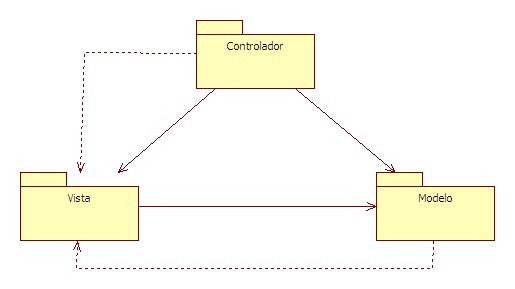
El Sistema SAM se desarrollará, a petición de los integrantes del LISI, bajo el patrón conocido como **MVC**, el cual dividirá nuestro software en tres (3) capas principales. Con el uso de dicho patrón se contará con un mejor entendimiento del sistema y facilitará las labores de desarrollo y mantenimiento de la aplicación.

Como se mencionó anteriormente, el patrón **MVC** se encarga de separar la aplicación en 3 capas, conocidas como el Modelo, la Vista y el Controlador.

El Modelo (también conocida como la capa de datos) es la capa en donde reside la información que maneja el sistema SAM. Estará conformado por un gestor de bases de datos (en nuestro caso, MySQL) para realizar el almacenamiento de datos. Esta capa deberá responder a las peticiones de información de estado de parte de la capa de Vista y a las instrucciones de modificación de estado, provenientes de la capa de Controlador.

La Vista (también conocida como la capa de aplicación) es la capa que se encarga de presentar la interfaz con el usuario. Contiene todo el código para generar la interfaz con el usuario, en lenguaje HTML, y se encarga tanto de mostrar la información del sistema como capturar los datos ingresados por los usuarios. Esta es la única capa que el usuario final llega a ver del sistema.

El Controlador (también conocida como la capa lógica) es la capa que se encarga de interactuar entre la capa de Vista y la capa de Modelo. Esta capa recibe peticiones del usuario a través de la capa de Vista y las transmite a la capa de Modelo. Luego, recibe los datos generados por la capa Modelo y se los hace llegar a la capa de Vista para que puedan ser mostradas la usuario. Este ciclo se repite tantas veces como el usuario genere peticiones (o acciones) en el sistema. Por último, esta capa es de suma importancia puesto que contiene toda la lógica del negocio y es quien se encarga de procesar los datos de entrada ingresados por los usuarios.



*Diagrama de Componentes del patrón MVC*

## Capas

### **Vista**

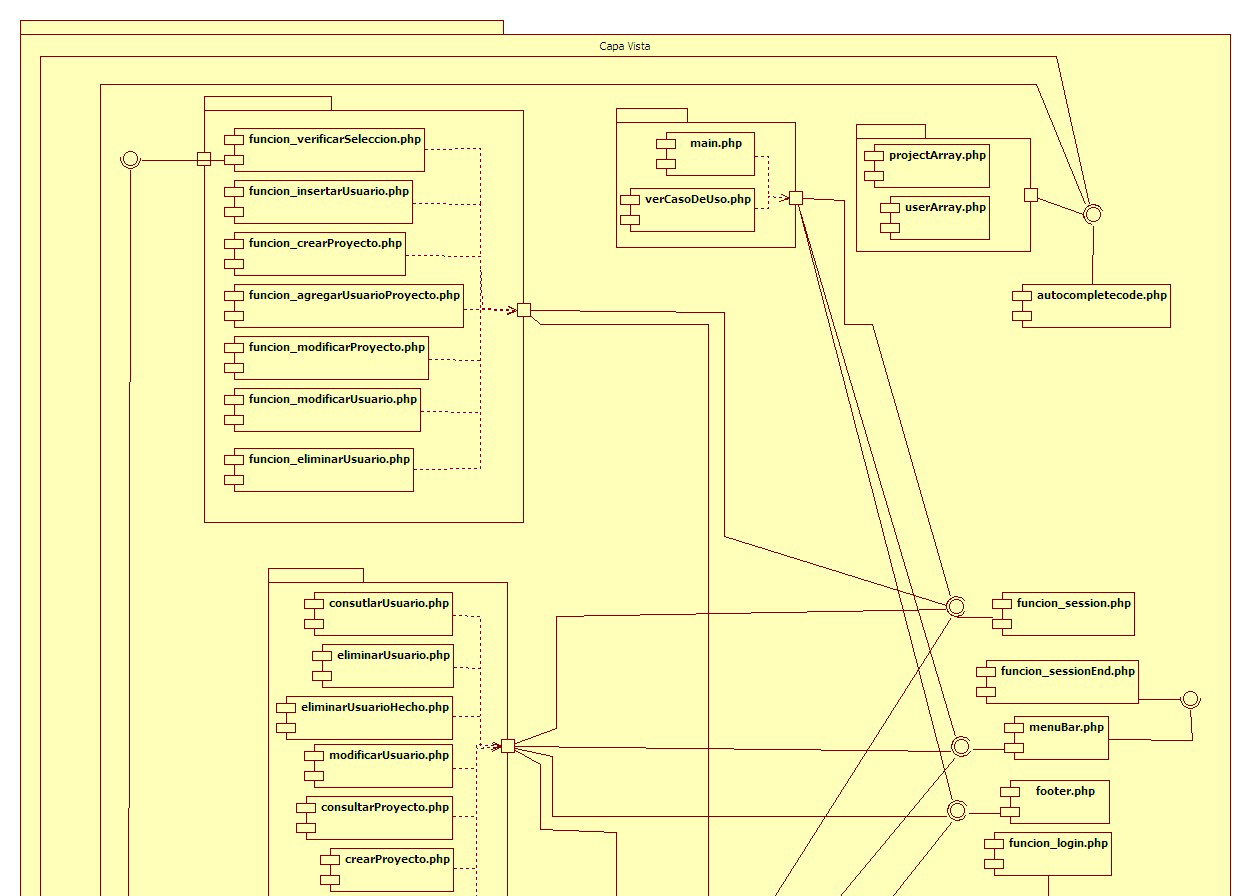
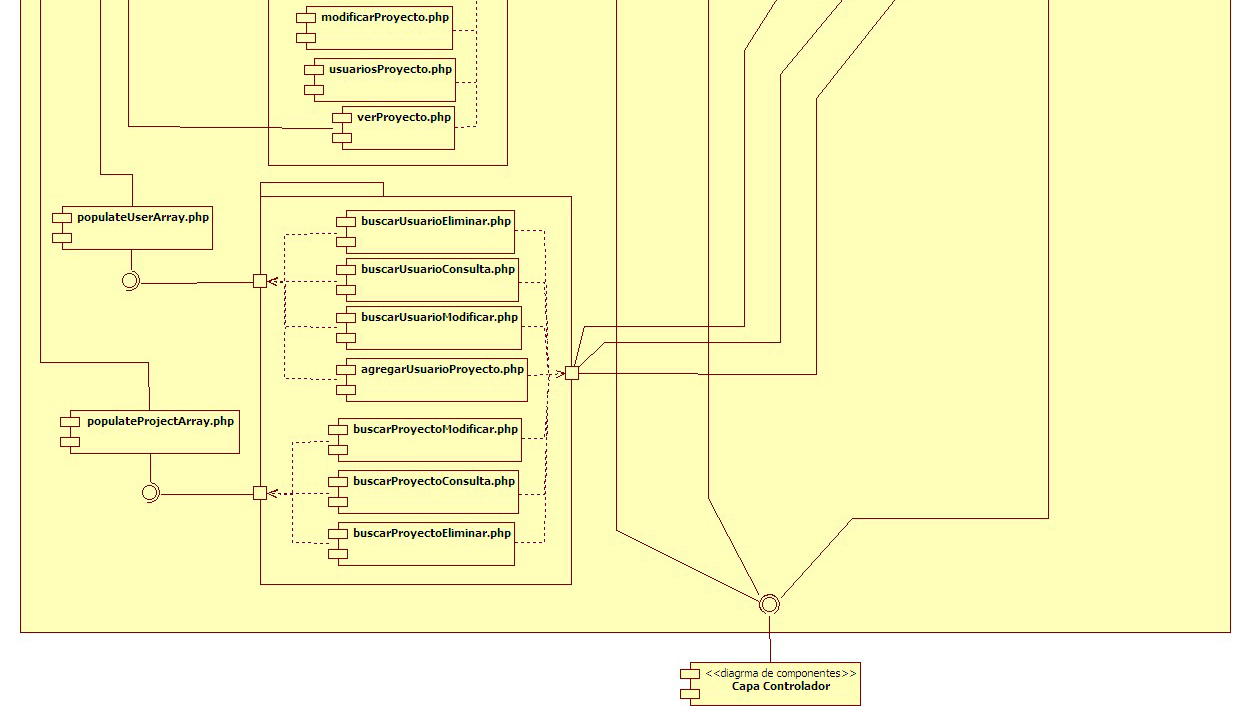
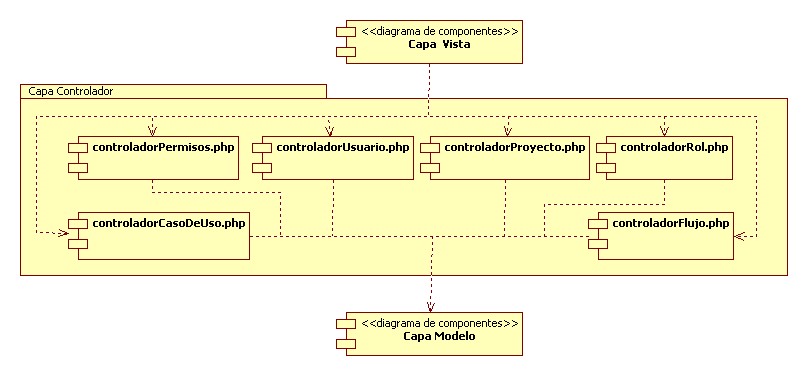


Diagrama de Componentes de la capa Vista (primera parte)



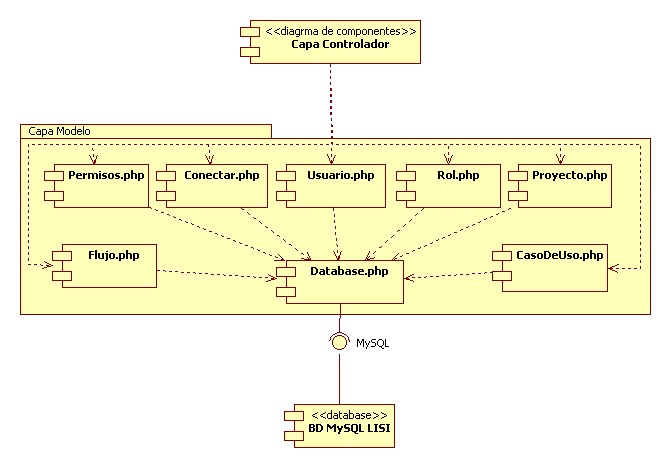
*Diagrama de Componentes de la capa Vista (segunda parte)*

### **Controlador**



*Diagrama de Componentes de la capa Controlador*

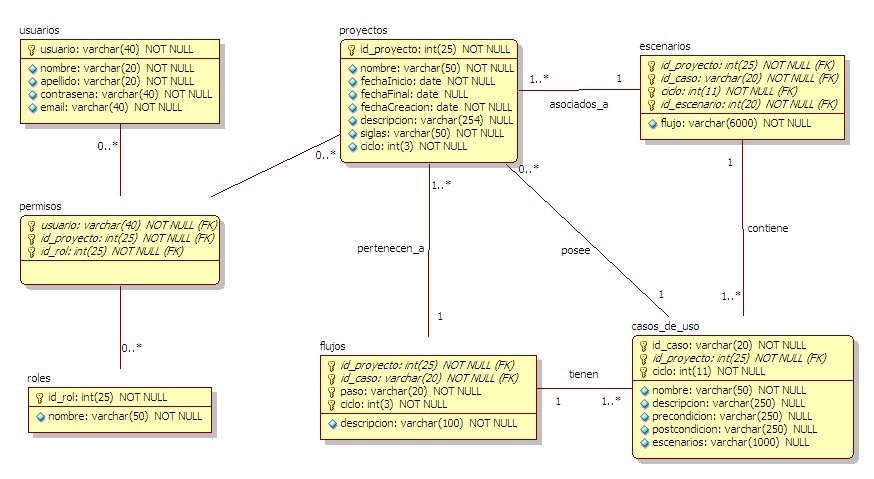
### **Modelo**



*Diagrama de Componentes de la capa Modelo*

# Vista de Datos

## Diagrama Entidad-Relación (ER)



## Diccionario de Datos

casos\_de\_uso

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atributo | Tipo | Null | Valor por Defecto | Descripción |
| id\_caso | varchar(20) | No | - | Identificador del CU. |
| nombre | varchar(50) | No | - | Indica el nombre del CU. |
| descripcion | varchar(250) | Si | NULL | Describe el CU. |
| precondiciones | varchar(250) | Si | NULL | Precondiciones que deben cumplirse antes de ejecutarse el CU. |
| postcondiciones | varchar(250) | Si | NULL | Postcondiciones que deben cumplirse después de ejecutarse el CU. |
| id\_proyecto | int(25) | No | - | Clave foránea que apunta a: proyectos → id\_proyecto. |
| ciclo | int(11) | No | - | Identifica el ciclo de pruebas. |
| escenarios | varchar(1000) | Si | NULL | Descripción de los escenarios asociados al CU. |

escenarios

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atributo | Tipo | Null | Valor por Defecto | Descripción |
| id\_proyecto | int(25) | No | - | Clave foránea que apunta a: proyectos → id\_proyecto. |
| id\_caso | varchar(20) | No | - | Clave foránea que apunta a: casos\_de\_uso → id\_caso. |
| ciclo | int(11) | No | - | Clave foránea que apunta a: casos\_de\_uso → ciclo. |
| id\_escenario | int(20) | No | - | Identificador de un escenario. |
| flujo | varchar(60000) | No | - | Descripción del flujo del escenario. |

flujos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atributo | Tipo | Null | Valor por Defecto | Descripción |
| id\_proyecto | int(25) | No | - | Clave foránea que apunta a: proyectos → id\_proyecto |
| id\_caso | varchar(20) | No | - | Clave foránea que apunta a: casos\_de\_uso → id\_caso |
| paso | varchar(20) | No | - | Titulo del paso. |
| descripcion | varchar(100) | No | - | Describe el paso. |
| ciclo | int(3) | No | - | Identificador del ciclo de pruebas. |

permisos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atributo | Tipo | Null | Valor por Defecto | Descripción |
| usuario | varchar(40) | No | - | Clave foránea que apunta a: usuarios → usuario |
| id\_proyecto | int(25) | No | - | Clave foránea que apunta a: proyectos → id\_proyecto |
| id\_rol | int(25) | No | - | Clave foránea que apunta a: roles → id\_rol |

proyectos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atributo | Tipo | Null | Valor por Defecto | Descripción |
| id\_proyecto | int(25) | No | - | Identifica un proyecto |
| nombre | varchar(50) | No | - | Indica el nombre del proyecto |
| fechaInicio | date | No | - | Indica la fecha de inicio del proyecto |
| fechaFinal | date | Yes | NULL | Indica la fecha de finalización del proyecto |
| fechaCreacion | date | No | - | Indica la fecha de creación del proyecto |
| descripcion | varchar(254) | Yes | NULL | La descripción del proyecto |
| siglas | varchar(50) | No | - | Las siglas del nombre del proyecto. |
| ciclo | int(3) | No | 0 | Identificador del ciclo de pruebas. |

Roles

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atributo | Tipo | Null | Valor por Defecto | Descripción |
| id\_rol | int(25) | No | - | Identificador del Rol |
| nombre | varchar(50) | No | - | Nombre del Rol |

Usuarios

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atributo | Tipo | Null | Valor por Defecto | Descripción |
| usuario | varchar(40) | No | - | Apodo del Usuario |
| nombre | varchar(20) | No | - | Nombre del Usuario |
| apellido | varchar(20) | No | - | Apellido del Usuario |
| contrasena | varchar(40) | No | - | Contraseña del Usuario en el sistema |
| email | varchar(40) | No | - | Dirección de e-mail del Usuario |

# Tamaño y Desempeño

# El Sistema SAM se puede catalogar como un sistema de pequeña envergadura, en vista de que sus clientes objetivos son las Pymes, las cuales cuentan (a lo sumo) con una cantidad de 100 empleados (de los cuales se puede asumir que el número de usuarios será estrictamente menor a esa cantidad). No se puede medir a priori el volumen de información que será cargada en el sistema debido a que esto dependerá de manera directamente proporcional a las actividades de la empresa cliente y el tamaño de los proyectos que ingresen en el sistema. La cantidad máxima de información que pueda alojarse en el sistema dependerá de la capacidad de alojamiento del servidor en el cual se implante el Sistema SAM.

El Sistema SAM también puede considerarse como un sistema de alta portabilidad, debido a sus bajos requerimientos de sistema: un servidor con soporte de PHP5 y MySQL, un explorador Web (Microsoft Internet Explorer o Mozilla Firefox) y una conexión de Internet (o Intranet). Adicionalmente, el patrón de arquitectura utilizado para su desarrollo garantiza reducir a un mínimo la complejidad del diseño de la arquitectura y maximizar la flexibilidad y mantenimiento del código. El sistema está diseñado para que cumpla acciones y peticiones en un periodo de 1 a 4 segundos, dado que se cuente con una buena velocidad de Internet/Intranet y que no ocurran fallas en el servidor de implantación.

El Sistema encuentra su mayor restricción en el estado de la conexión de Internet/Intranet entre la máquina cliente y el servidor que aloja al sistema. Una baja velocidad de conexión en la red puede afectar los tiempos de respuesta del sistema o, en su peor caso, impedir hacer uso del sistema.

# Calidad

|  |  |  |  |
| --- | --- | --- | --- |
| **Requerimiento** | **Diagrama** | **Elemento** | **Observaciones** |
| Tolerancia a fallos | Diagrama de Implantación Estándar del Sistema SAM | Servidor de Respaldo | Se recomienda el uso de un servidor de respaldo que monitoree el estado del servidor de aplicación principal, con el fin de manejar aquellas fallas que puedan ocurrir en el mismo. |
| Uso de protocolos de comunicación pertenecientes a entornos Web. | Diagrama de Implantación Estándar del Sistema SAM | Enlaces “Maquina Cliente – Servidor de Aplicación” y “Servidor de Aplicación – Servidor de Prueba” | - |
| Lenguajes, Componentes Comprados y Licenciamiento | Diagrama de Implantación Estándar del Sistema SAM | Servidor de Aplicación | Como se puede ver en el diagrama, los componentes del software fueron desarrollados con PHP, JavaScript, CSS, los cuales son lenguajes de software libre. Adicionalmente, la base de datos utilizada es MySQL, la cual también es software libre. |
| Arquitectura MVC | Diagrama de Componentes del Patrón MVC y Diagrama de Componentes de las Capas Vista, Controlador y Modelo | - | A pesar de que no se está utilizando un framework para el desarrollo de la aplicación, se está haciendo uso del patrón MVC durante el desarrollo del mismo. |

# 11 Arquitectura tecnológica de la aplicación

(Colocar el diseño/gráfico de la arquitectura tecnológica de la aplicación)