

Sistema de Monitoreo y Simulación de Tráfico por Semáforos

Sistema de Monitoreo y Simulación de Tráfico por Semáforos	1
1. Introducción	1
2. Estructura del Proyecto	1
2.1 Diagrama de clases	3
2.2 Diagrama de paquetes	3
2.3 Arquitectura del sistema	3
2.4 Diagrama de actividades	4
3. Tecnologías Utilizadas	4
4. Instalación y Configuración	4
4.1 Requisitos Previos	4
4.2 Configuración del Servidor	4
5. Descripción de Módulos	5
5.1 Controladores (Controller/):	5
5.2 Modelos (Model/):	5
5.3 Vistas (View/):	5
6. Seguridad	5
7. Mantenimiento y Actualizaciones	6
8. Contacto y Soporte	6

1. Introducción

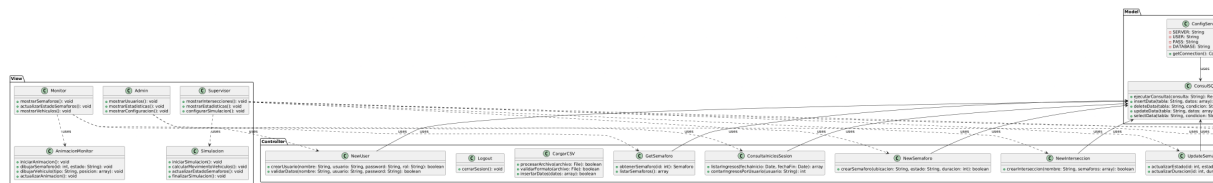
Este documento proporciona la información técnica necesaria para la instalación, configuración y mantenimiento del sistema de monitoreo y simulación de tráfico basado en sensores de semáforos. El sistema está diseñado para modelar intersecciones urbanas y evaluar el flujo vehicular en diferentes escenarios.

2. Estructura del Proyecto

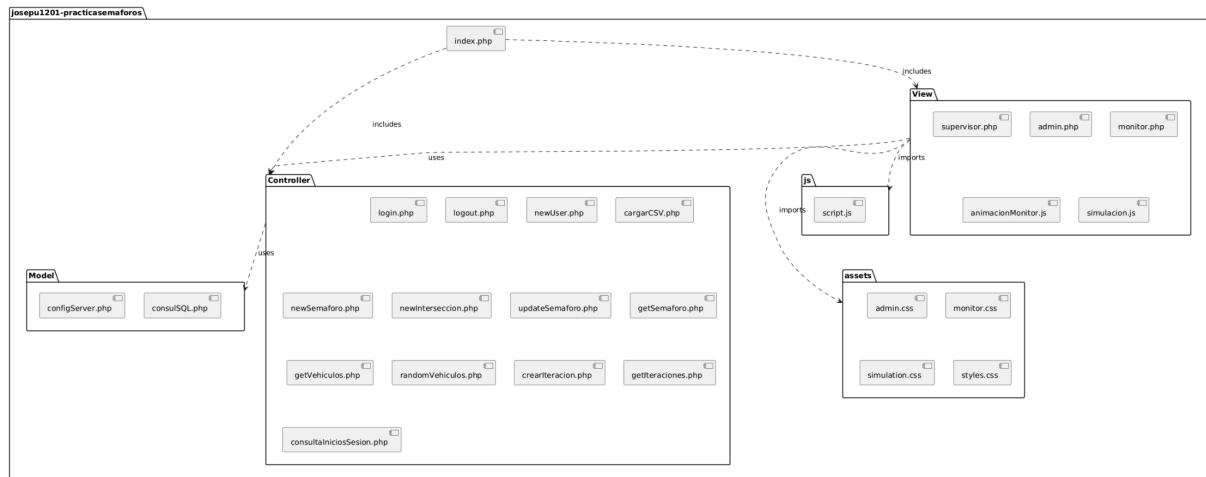
El proyecto está organizado con la siguiente estructura de directorios:

```
└─ josepu1201-practicasemaforos/
   └─ index.php
   └─ assets/
      └─ admin.css
      └─ monitor.css
      └─ simulation.css
      └─ styles.css
   └─ Controller/
      └─ cargarCSV.php
      └─ consultaIniciosSesion.php
      └─ crearIteracion.php
      └─ getIteraciones.php
      └─ getSemaforo.php
      └─ getVehiculos.php
      └─ login.php
      └─ logout.php
      └─ newInterseccion.php
      └─ newSemaforo.php
      └─ newUser.php
      └─ randomVeiculos.php
      └─ updateSemaforo.php
   └─ js/
      └─ script.js
   └─ Model/
      └─ configServer.php
      └─ consulSQL.php
   └─ View/
      └─ admin.php
      └─ animacionMonitor.js
      └─ monitor.php
      └─ simulacion.js
      └─ supervisor.php
```

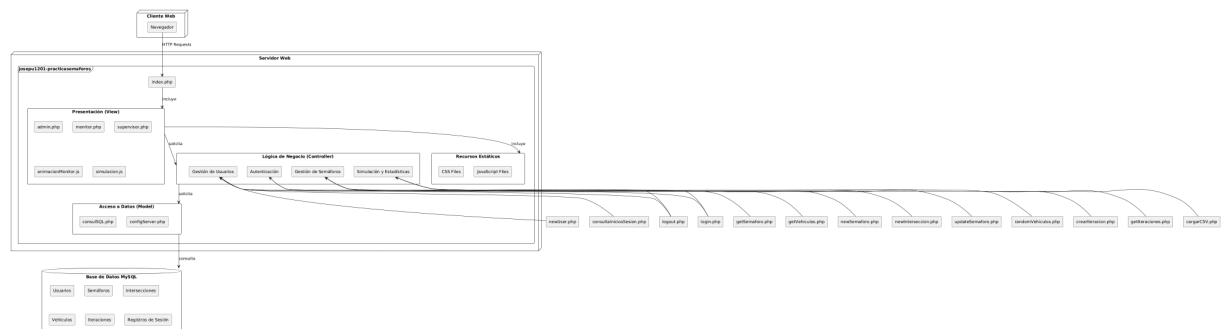
2.1 Diagrama de clases



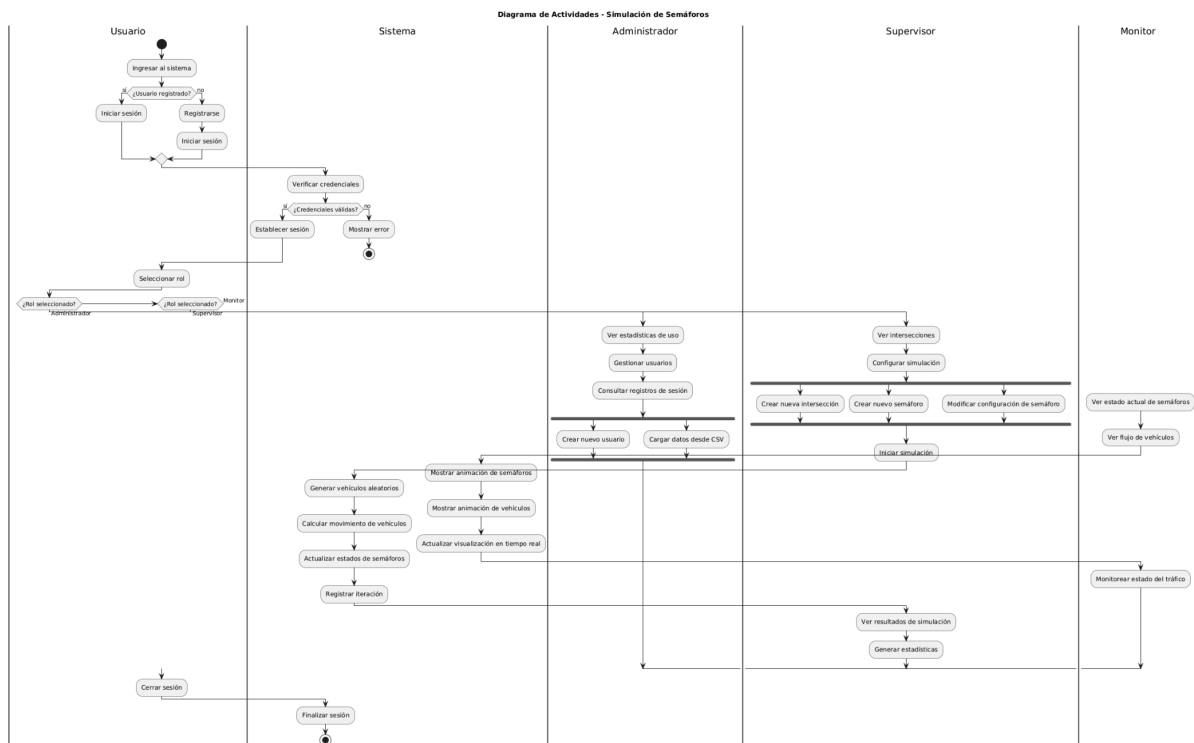
2.2 Diagrama de paquetes



2.3 Arquitectura del sistema



2.4 Diagrama de actividades



3. Tecnologías Utilizadas

- **Backend:** PHP 8 con Apache y MySQL.
- **Frontend:** HTML, CSS, JavaScript (Vanilla y Bootstrap para diseño responsivo).
- **Base de Datos:** MySQL.
- **Frameworks y Librerías:** Bootstrap para el diseño visual, jQuery para manipulación del DOM.
- **Arquitectura:** MVC (Modelo-Vista-Controlador).

4. Instalación y Configuración

4.1 Requisitos Previos

- Servidor web Apache.
- PHP 8 o superior.
- MySQL 5.7 o superior.
- XAMPP o equivalente para desarrollo local.

4.2 Configuración del Servidor

1. Descargar y extraer el código fuente en el directorio del servidor.
2. Configurar la base de datos ejecutando el script SQL ubicado en `Model/consulSQL.php`.
3. Editar `Model/configServer.php` y ajustar las credenciales de MySQL.
4. Configurar Apache para apuntar al directorio del proyecto.

5. Descripción de Módulos

5.1 Controladores (Controller/):

- `login.php` y `logout.php`: Manejan la autenticación de usuarios.
- `getSemaforo.php`: Obtiene el estado de los semáforos.
- `updateSemaforo.php`: Permite modificar los tiempos de los semáforos.
- `getVehiculos.php`: Recupera información de los vehículos en simulación.
- `randomVeiculos.php`: Genera vehículos aleatorios para la simulación.
- `newInterseccion.php`: Agrega nuevas intersecciones al sistema.

5.2 Modelos (Model/):

- `configServer.php`: Configuración de conexión a la base de datos.
- `consulSQL.php`: Consulta y manipulación de datos en MySQL.

5.3 Vistas (View/):

- `admin.php`: Interfaz para administradores.
- `monitor.php`: Vista de monitoreo en tiempo real.
- `supervisor.php`: Panel de supervisión y generación de reportes.
- `simulacion.js`: Script para manejar la animación de la simulación de tráfico.

6. Seguridad

- Autenticación basada en sesiones.
- Validación y sanitización de entradas para prevenir inyecciones SQL y XSS.
- Control de acceso basado en roles (Administrador, Monitor, Supervisor).
- Hashing de contraseñas con `bcrypt`.

7. Mantenimiento y Actualizaciones

- **Respaldo de la base de datos:** Se recomienda programar backups automáticos.
- **Actualización del sistema:** Se debe probar en un entorno de desarrollo antes de actualizar en producción.
- **Monitoreo de logs:** Revisar `logs/error.log` para detectar posibles fallos.

8. Contacto y Soporte

Para soporte técnico, contactar con el equipo de desarrollo a través de correo electrónico o plataforma de gestión de incidencias.

Jose Pu: <https://github.com/JosePu1201>