

INFORME DE LAS PRÁCTICAS DE EXPERIMENTACIÓN Y APLICACIÓN DE LOS APRENDIZAJES

1. Datos Informativos:

Facultad:	Ciencias Administrativas, Gestión Empresarial e Informática
Carrera:	Software
Asignatura:	Arquitectura de Software
Ciclo:	5to
Docente:	Ing. Carrión Buenaño Darwin Paúl
Título de la práctica:	Vista arquitectónica del sistema de georreferenciación de la cooperativa de transporte Universidad de Bolívar de la ciudad de Guaranda
No. de práctica:	1
Escenario o ambiente de aprendizaje de la practica	Laboratorio
No. de horas:	24
Fecha:	16 de octubre del 2025
Estudiantes:	Valentín Valle Carlos Caiza Alex Chisag
Calificación	

2. Introducción:

La arquitectura de software es el pilar fundamental en el desarrollo de sistemas complejos, ya que define su estructura, componentes y las interacciones entre ellos. Para documentar y comunicar esta estructura de manera efectiva, se utilizan modelos de vistas arquitectónicas. El modelo "4+1" de Philippe Kruchten es un estándar de la industria que permite describir un

sistema desde cinco perspectivas distintas, asegurando que se cubran los intereses de todos los involucrados (stakeholders), desde los desarrolladores hasta los usuarios finales.

Este informe aplica el modelo de vistas arquitectónicas para documentar el diseño del sistema web **"Vida Activa UEB"**, una aplicación orientada a la planificación y seguimiento de rutinas de ejercicio para los estudiantes de la Universidad Estatal de Bolívar. A través de las vistas Lógica, de Desarrollo, de Procesos y Física, se detallará la estructura interna y el comportamiento del software.

3. **Objetivo de la práctica:**

Aplicar el modelo de vistas arquitectónicas "4+1" para analizar, documentar y comunicar de manera integral la arquitectura del sistema web "Vida Activa UEB", asegurando que el diseño sea comprensible, robusto y alineado con los requisitos del proyecto.

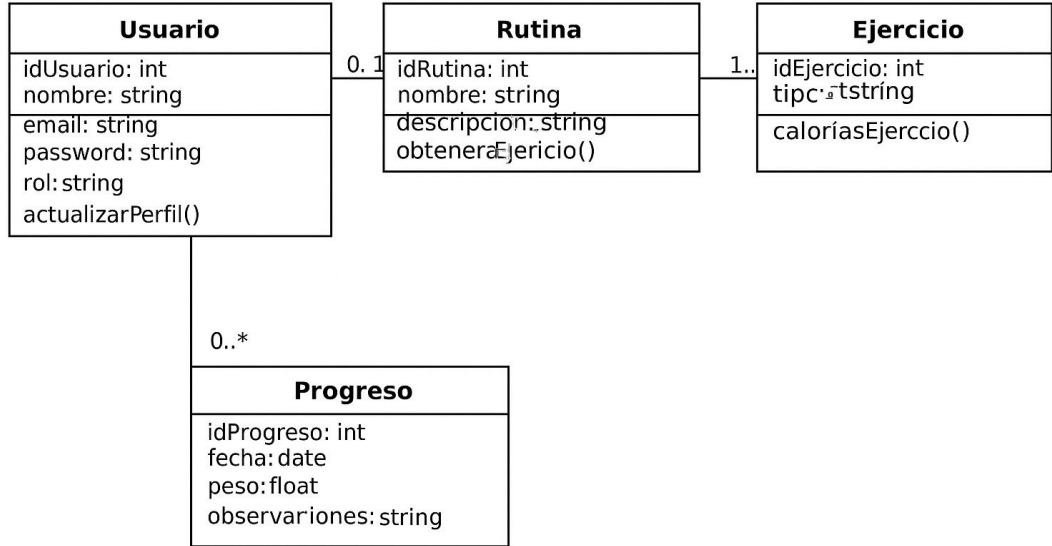
4. **Descripción del desarrollo de la práctica:**

Para documentar la arquitectura del sistema "Vida Activa UEB", se aplicó el modelo 4+1 de Kruchten. Este modelo descompone el diseño del software en cuatro vistas principales que describen diferentes aspectos del sistema, unificadas por una vista de escenarios.

4.1. Vista Lógica

Esta vista se enfoca en la funcionalidad que el sistema provee a los usuarios finales. Describe la estructura de los datos y los componentes de negocio. Para "Vida Activa UEB", esta vista se representa mediante un Diagrama de Clases, que es el equivalente UML de un Diagrama Entidad-Relación. Muestra las clases principales del sistema (Usuarios, Rutinas, Ejercicios, etc.), sus atributos (columnas de la base de datos) y las relaciones entre ellas.

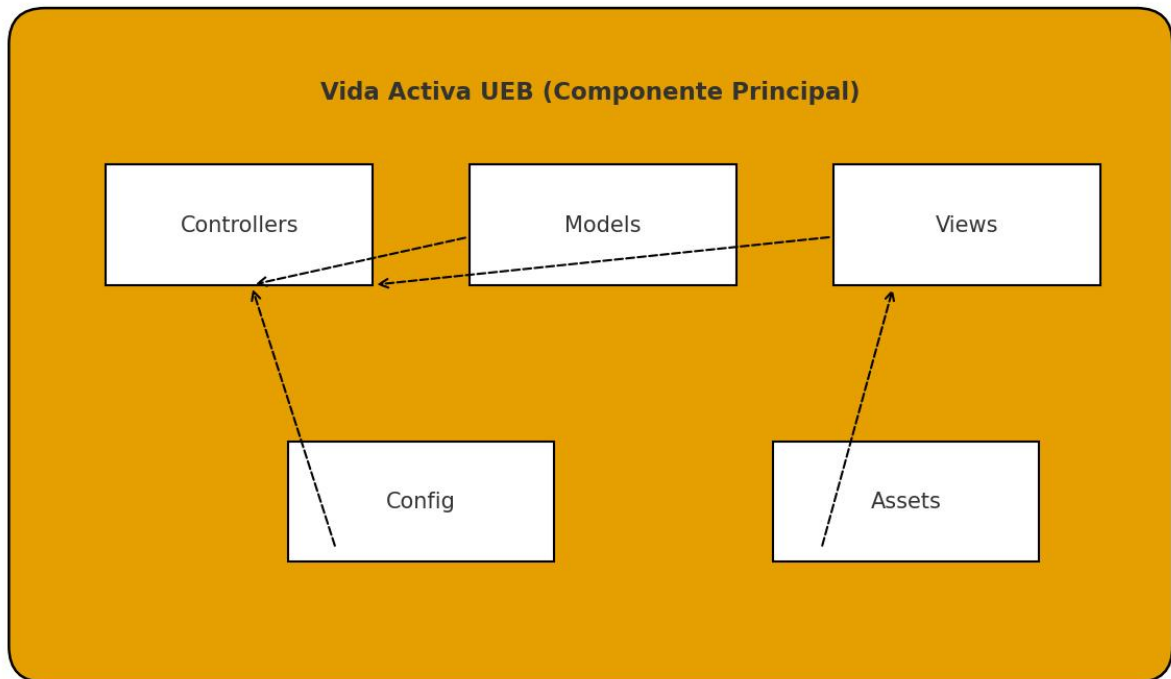
Diagrama de Clases - Vista Lógica



4.2. Vista de Desarrollo

Esta vista, también conocida como vista de implementación, ilustra cómo está organizado el código fuente. Su objetivo es facilitar el trabajo de los desarrolladores. En nuestro proyecto, hemos implementado el patrón arquitectónico Modelo-Vista-Controlador (MVC). El Diagrama de Componentes muestra esta organización, dividiendo el sistema en subsistemas lógicos (models, views, controllers, config, assets) y las dependencias que existen entre ellos.

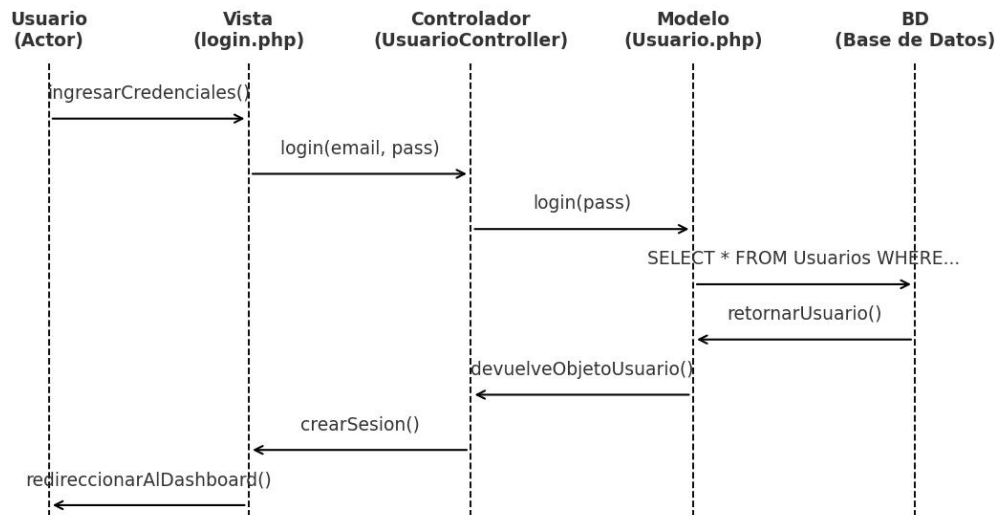
Diagrama de Componentes - Vista de Desarrollo



4.3. Vista de Procesos

La vista de procesos describe el comportamiento dinámico del sistema, es decir, cómo los componentes interactúan en tiempo de ejecución. Se enfoca en la concurrencia y la comunicación entre objetos. Para ilustrar esto, se utiliza un Diagrama de Secuencia que modela el flujo del proceso de inicio de sesión de un usuario. Este diagrama muestra paso a paso cómo la vista, el controlador y el modelo colaboran para autenticar al usuario contra la base de datos.

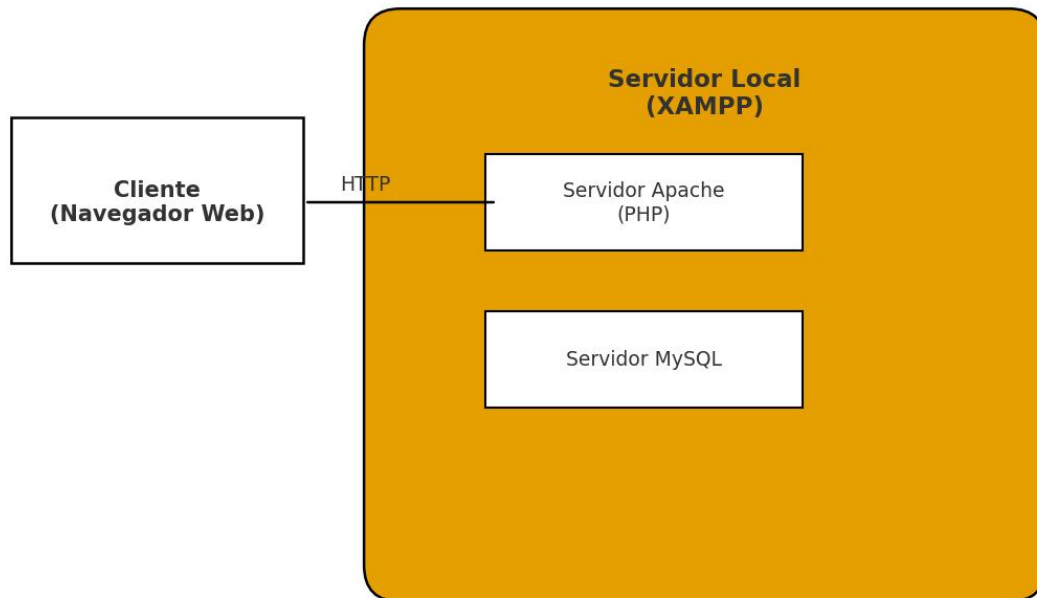
Diagrama de Secuencia - Login



4.4. Vista Física

Esta vista describe cómo se despliega el software en la infraestructura de hardware. Para "Vida Activa UEB", el despliegue se realiza en un entorno de servidor local simulado con XAMPP. El Diagrama de Despliegue muestra los nodos físicos involucrados: el Cliente (navegador web del usuario) y el Servidor (que ejecuta Apache para procesar PHP y MySQL para la base de datos), y cómo se comunican a través del protocolo HTTP.

Diagrama de Despliegue - Vista Física



4.5. Vista de Escenarios (La "+1")

Esta vista no tiene un diagrama propio, sino que sirve para unificar las otras cuatro. Consiste en los casos de uso principales del sistema (ej. "Registrar un nuevo usuario", "Crear una rutina personalizada") que ilustran y validan que el diseño arquitectónico plasmado en las otras cuatro vistas funciona en conjunto para cumplir con los requisitos.

5. Metodología:

El desarrollo de la práctica se llevó a cabo siguiendo los siguientes pasos:

Análisis del Modelo 4+1: Se estudiaron los fundamentos teóricos de cada una de las cinco vistas arquitectónicas.

Aplicación al Proyecto: Se analizó el sistema "Vida Activa UEB" para identificar los elementos correspondientes a cada vista (clases, componentes, procesos y nodos de hardware).

Diseño de Diagramas UML: Utilizando la herramienta de modelado Lucidchart, se crearon los cuatro diagramas UML (Clases, Componentes, Secuencia y Despliegue) que representan visualmente cada vista.

Documentación: Se redactó el presente informe, describiendo cada vista, explicando su propósito y detallando su implementación en el contexto del proyecto.

6. Resultados obtenidos:

Como resultado de la aplicación del modelo 4+1, se obtuvo una documentación arquitectónica completa y multifacética del sistema "Vida Activa UEB". Los entregables concretos de esta práctica son:

- Un **Diagrama de Clases** que modela la estructura estática de los datos (Vista Lógica).
- Un **Diagrama de Componentes** que ilustra la organización del código fuente bajo el patrón MVC (Vista de Desarrollo).
- Un **Diagrama de Secuencia** que detalla el flujo dinámico del proceso de inicio de sesión (Vista de Procesos).
- Un **Diagrama de Despliegue** que representa la infraestructura de hardware sobre la cual se ejecuta la aplicación (Vista Física).

Estos diagramas, en conjunto con su descripción textual, proporcionan una visión integral del diseño del sistema, sirviendo como una guía fundamental para el equipo de desarrollo y futuros mantenimientos.

7. Conclusiones:

- El uso del modelo de vistas arquitectónicas 4+1 demostró ser una herramienta eficaz para descomponer la complejidad del sistema "Vida Activa UEB" en partes más comprensibles, facilitando el análisis y la comunicación del diseño.
- La separación en vistas (Lógica, Desarrollo, Procesos y Física) permitió abordar el diseño desde todas las perspectivas relevantes, garantizando que tanto los requisitos funcionales como los no funcionales fueran considerados en la arquitectura.
- La documentación generada no solo sirve como un registro del diseño actual, sino también como una base sólida para la evolución futura del proyecto, permitiendo que nuevos desarrolladores comprendan rápidamente la estructura y el funcionamiento del software.

8. Recomendaciones:

- Se recomienda mantener esta documentación arquitectónica actualizada a medida que el proyecto evolucione. Cualquier cambio significativo en la base de datos o en la lógica de la aplicación debe reflejarse en los diagramas correspondientes.
- Utilizar los diagramas como la guía principal durante la fase de implementación de nuevas funcionalidades, asegurando que el equipo de desarrollo se adhiera a la estructura y los patrones ya establecidos.
- Para futuras prácticas, se podría ampliar la Vista de Procesos detallando otros flujos críticos del sistema, como la "Creación de una rutina personalizada", para tener una guía aún más completa.

9. Bibliografía:

GUÍA DE PRÁCTICAS DE APLICACIÓN Y/O EXPERIMENTACIÓN DE LOS APRENDIZAJES (PAE O APE). Asignatura Arquitectura de Software. (Documento proporcionado por el docente).

Kruchten, P. B. (1995). The 4+1 View Model of Architecture. IEEE Software.

10. Anexos:

4.2. Vista de Desarrollo

Esta vista, también conocida como vista de implementación, ilustra cómo está organizado el código fuente. Su objetivo es facilitar el trabajo de los desarrolladores. En nuestro proyecto, hemos implementado el patrón arquitectónico Modelo-Vista-Controlador (MVC). El Diagrama de Componentes muestra esta organización, dividiendo el sistema en subsistemas lógicos (models, views, controllers, config, assets) y las dependencias que existen entre ellos.

(Aquí es donde tu compañero debe insertar el Diagrama de Componentes)

4.3. Vista de Procesos

La vista de procesos describe el comportamiento dinámico del sistema, es decir, cómo los componentes interactúan en tiempo de ejecución. Se enfoca en la concurrencia y la comunicación entre objetos. Para ilustrar esto, se utiliza un Diagrama de Secuencia que modela el flujo del proceso de inicio de sesión de un usuario. Este diagrama muestra paso a paso cómo la vista, el controlador y el modelo colaboran para autenticar al usuario contra la base de datos.

(Aquí es donde tu compañero debe insertar el Diagrama de Secuencia del Login)

4.4. Vista Física

Esta vista describe cómo se despliega el software en la infraestructura de hardware. Para "Vida Activa UEB", el despliegue se realiza en un entorno de servidor local simulado con XAMPP. El Diagrama de Despliegue muestra los nodos físicos involucrados: el Cliente (navegador web del usuario) y el Servidor (que ejecuta Apache para procesar PHP y MySQL para la base de datos), y