

# Práctica: Definición de la Arquitectura de Software

## Fase 1: Introducción a la Arquitectura de Software

La arquitectura de software es la estructura fundamental de un sistema, que comprende los elementos del software, sus propiedades visibles externamente y las relaciones entre ellos. Define el "esqueleto" organizativo que asegura que el sistema cumpla con los requisitos técnicos y de negocio, abarcando decisiones críticas sobre componentes, capas y patrones de diseño (como MVC o Cliente-Servidor) que impactan directamente en la calidad, mantenibilidad y escalabilidad del producto final.

## Fase 2: Actividad Práctica (Proyecto AlgeVisual)

### 1. Identificación de Requisitos Clave

Para definir la arquitectura de **AlgeVisual**, se consideraron los siguientes requisitos del sistema:

- **Accesibilidad Multiplataforma (RNF-01 y RNF-05):** La aplicación debe funcionar en dispositivos móviles (Android/iOS) y permitir acceso a docentes vía web.
- **Centralización de Datos (RF-03 y RF-07):** El progreso de los estudiantes debe guardarse en una base de datos central para que los docentes puedan consultarlo en tiempo real desde su propio panel.
- **Escalabilidad:** Se requiere capacidad para agregar nuevos módulos de ejercicios o juegos en el futuro sin reestructurar todo el sistema.

### 2. Estilo Arquitectónico Seleccionado

Se seleccionó el estilo **Cliente-Servidor** (implementado mediante una arquitectura de N-Capas con API REST).

### 3. Diseño de Arquitectura (Componentes Principales)

La estructura se divide en tres capas lógicas:

- **Capa de Presentación (Front-end / Clientes):**
  - **App Móvil (Estudiante):** Desarrollada para Android/iOS. Se encarga de la interacción visual, los ejercicios interactivos y la síntesis de voz (accesibilidad).
  - **Portal Web (Docente):** Interfaz web para la gestión de grupos y revisión de estadísticas.

- **Capa de Lógica de Negocio (Back-end / Servidor):**
  - **API REST:** Actúa como intermediario. Recibe las solicitudes de los clientes (ej. "Validar respuesta", "Obtener lista de alumnos"). Aquí reside la lógica matemática y de seguridad.
  
- **Capa de Datos (Base de Datos):**
  - **Servidor de Base de Datos:** Almacena de forma persistente los usuarios, ejercicios, historial de calificaciones y configuraciones.

#### 4. Justificación de las Decisiones

Se eligió la arquitectura **Cliente-Servidor** porque satisface la necesidad de **sincronización** entre distintos dispositivos (requisito RNF-01). Al separar la lógica (API) de la interfaz (App/Web), garantizamos que si un alumno cambia de celular, su progreso (RF-03) se mantiene intacto en el servidor. Además, facilita el mantenimiento: si necesitamos corregir una fórmula matemática, solo actualizamos el servidor sin necesidad de obligar a todos los usuarios a actualizar la app en sus tiendas.