

Trabajo Práctico 1 - Parte 2

Aplicación de la Arquitectura al Proyecto

Objetivo

Aplicar los conceptos de arquitectura vistos en clase (estilos, patrones, vistas, aspectos transversales y atributos de calidad) al proyecto iniciado en Ingeniería de Software I.

3.1. Atributos de calidad y escenarios

Los atributos calidad que consideramos críticos para nuestro proyecto luego de la votación de escenarios de calidad, fueron:

1. Disponibilidad
2. Interoperabilidad
3. Usabilidad

Los escenarios de calidad para cada uno de ellos fueron (especificando estímulo, respuesta y medida de la respuesta):

Disponibilidad

El sistema debe estar disponible para consultas de contenido las 24 horas, incluso en días feriados.	Acceso fuera de horario laboral	El acceso es posible sin restricciones	99% de disponibilidad anual
--	---------------------------------	--	-----------------------------

Interoperabilidad

Correcta visualización de contenido multimedia	Ingreso de videos cargados en la página se obtienen de distintas fuentes	Se observan los videos igualmente sin ningún problema sobre su origen	100% de integraciones correctas
--	--	---	---------------------------------

Usabilidad

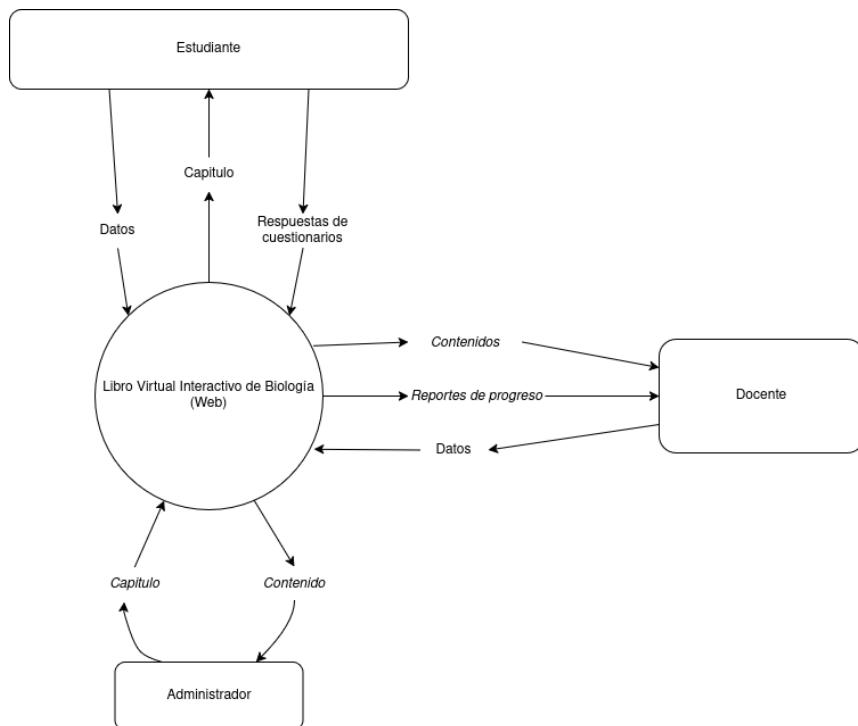
Un alumno accede a la plataforma desde una tablet y navega por los contenidos y cuestionarios.	Uso desde dispositivo móvil	La interfaz se adapta correctamente y permite completar todas las actividades	sin inconvenientes
--	-----------------------------	---	--------------------

3.2. Vistas de arquitectura

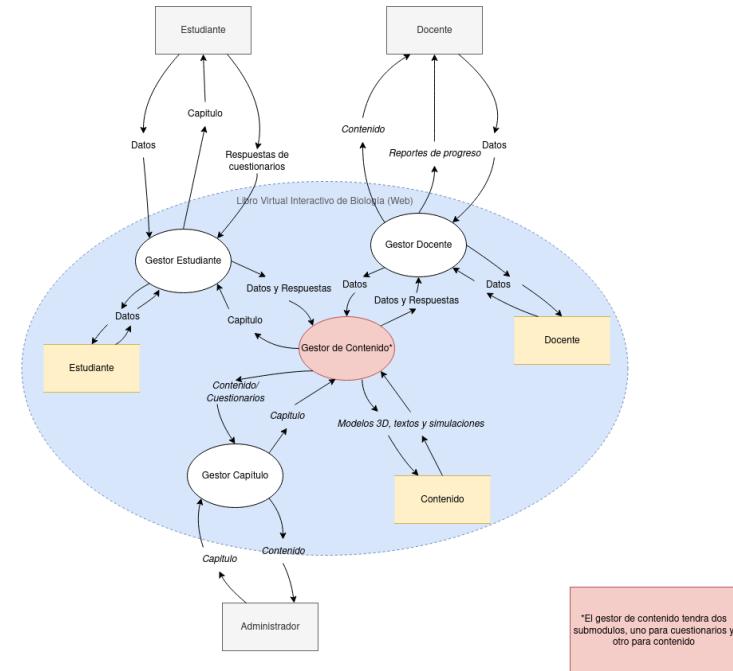
Utilizando el modelo de vistas C4 la arquitectura de nuestro proyecto es:

Contexto del sistema

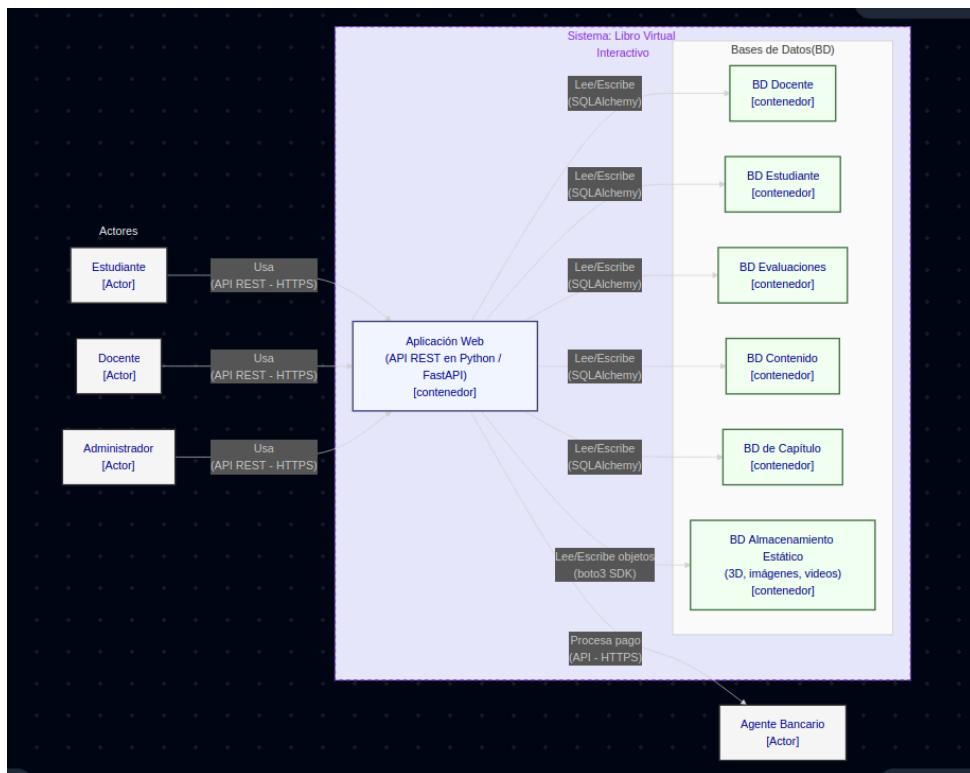
- Diagrama nivel 0:



- Diagrama nivel 1:

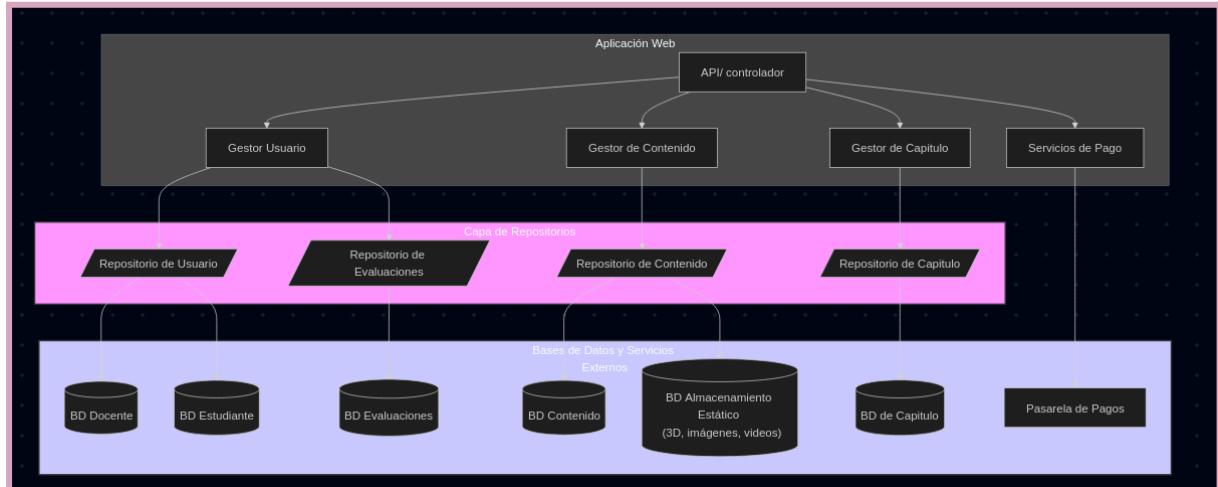


Contenedores



[Diagrama contenedores](#)

Componentes



[link diagrama comp](#)

Todo pertenece al mismo cuadro, es la ampliación del sistema libro virtual que se observa en diagrama de contenedores

3.3. Aspectos transversales

Los aspectos transversales que consideramos deben incluirse en el proyecto son:

- Autenticación y autorización: necesarios porque existen distintos tipos de usuarios (alumnos, docentes, administradores, premium). Impactan en la capa de presentación (restricción de vistas) y en los servicios (validación de permisos).
- Auditoría: asegura trazabilidad de accesos y modificaciones de contenidos, clave en un entorno educativo y con pagos premium. Impacta en datos (logs) y servicios (registro de eventos).
- Gestión de roles y perfiles: complementa autenticación/autorización, simplifica la administración de permisos. Impacta en la lógica de aplicación.
- Gestión de excepciones: mejora la robustez y evita pérdida de información (ej. envío de cuestionarios). Impacta en backend y en la UX.
- Instrumentación: permite monitorear rendimiento, tiempos de respuesta y uso de recursos, fundamental por el uso de multimedia y simulaciones 3D. Impacta en infraestructura y servicios.

- Interoperabilidad: necesaria para exportar reportes y resultados en formatos estándar, y para futuras integraciones con otros sistemas educativos. Impacta en servicios de aplicación y APIs.
- Mailing y notificaciones: mantiene la comunicación activa entre sistema y usuarios (recordatorios, resultados). Impacta en backend (servicio de notificaciones) y en la capa de infraestructura (proveedor de correo).

Conclusiones

La decisión arquitectónica que más nos costó fue aplicar correctamente el modelo C4 a nuestro proyecto, sobre todo los diagramas de contenedores y componentes. Fue difícil entender cómo pasar de la descripción general del sistema a los diferentes niveles de detalle (contexto, contenedores, componentes, código) sin repetir información o confundir roles. Por lo que se discutió sobre qué debía ir en cada nivel y cómo representar de forma clara la relación entre los actores externos, los contenedores y los módulos internos de la aplicación.