

# Ejercicio de Aplicación

Todos los temas vistos hasta el momento, incluyendo Integrabilidad, Deployabilidad, y patrones asociados

## Mecánica de trabajo

Cada equipo trabaja sobre el ejercicio en forma simultánea.

El equipo luego de leer la letra deberá y realizar dos actividades, una mediante Intedashboard y otra utilizando el repositorio correspondiente en la ORG del curso:

## Intedashboard

- Identificar los atributos de calidad y restricciones relevantes.
- Responder el **primer cuestionario** en intedashboard sobre el ejercicio y conceptos asociados.
- Luego de haber trabajado en la primera versión del ADR **responda las últimas dos preguntas del segundo cuestionario en Intedashboard.**

## Utilizar el repositorio para

- Comunicar el diseño utilizando las ADRs y UML donde se visualicen mejor los mecanismos seleccionados.
- **Al finalizar la clase realicen un commit y push a main** para marcar lo avanzado hasta el momento.
- **Luego pueden continuar mejorando su solución hasta la media noche del día siguiente, momento en el cual deberán realizar una PR asignada al docente y revisando contra el checklist asociado.**

---

Una empresa de tecnología está evaluando construir una solución de software para organizar reuniones, la cual quiere vender como servicio. El arquitecto jefe de la empresa elaboró cuatro opciones de solución – a muy alto nivel - las cuales se presentan más adelante. El objetivo del arquitecto fue presentar, utilizando parte de los elementos de la solución, las distintas arquitecturas que consideró.

Antes de continuar con la construcción de la plataforma, la empresa contrató los servicios de su equipo para que recomiende la mejor opción a implementar.

Las necesidades y restricciones planteo la empresa fueron las siguientes:

1. La solución va a ser desarrollada por equipos independientes en los que participan especialistas con conocimiento específico de las áreas del negocio que involucra la solución, Ej. Registro, Planificación de reuniones, manejo de usuarios y clientes, etc.
2. Los equipos de desarrollo son pequeños (7 a 12 personas), integrado por personas que en su mayoría son junior Software Engineers, con muy poca experiencia en desarrollo y diseño.
3. Se espera poder evaluar el éxito de la solución una vez puesta en producción.

4. Se espera poder desarrollar y probar con grupos de usuarios, en producción, nuevas features lo más rápido que se pueda y al menor costo posible.
5. Si todo sale de acuerdo con el plan estratégico del producto, se espera que la solución, a mediano plazo, cuente con una base de usuarios dentro de los cientos de miles y que pueda mantener una buena latencia en las funcionalidades de Reuniones y Registro.

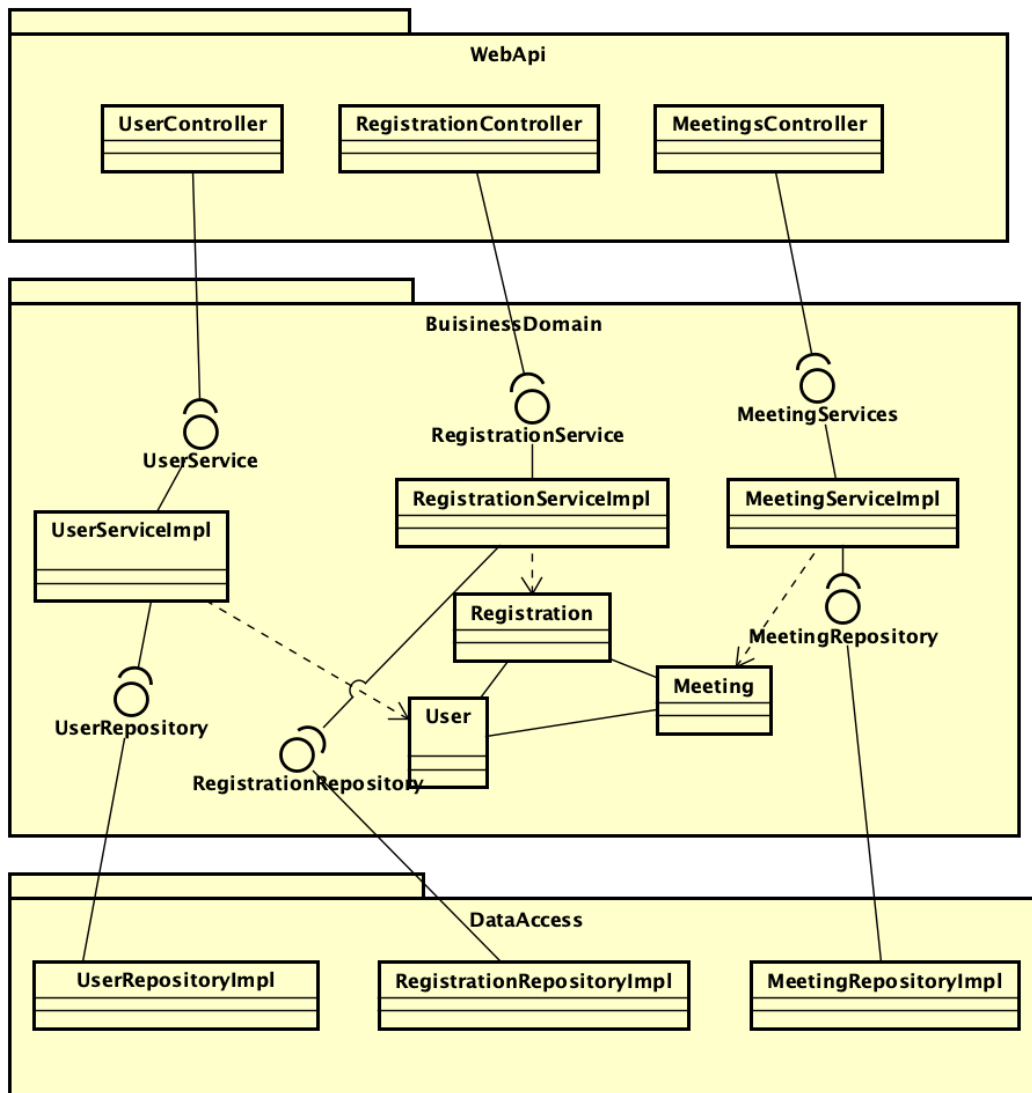
Se le pide que evalúe las arquitecturas con base en las necesidades planteadas y que elabore un ADR justificando su solución y evaluando las ventajas y desventajas respecto a su segunda opción.

## Opciones arquitectónicas

En todas las opciones de arquitecturas se encuentran una serie de módulos y componentes similares, para los cuales se describe sus responsabilidades a muy alto nivel.

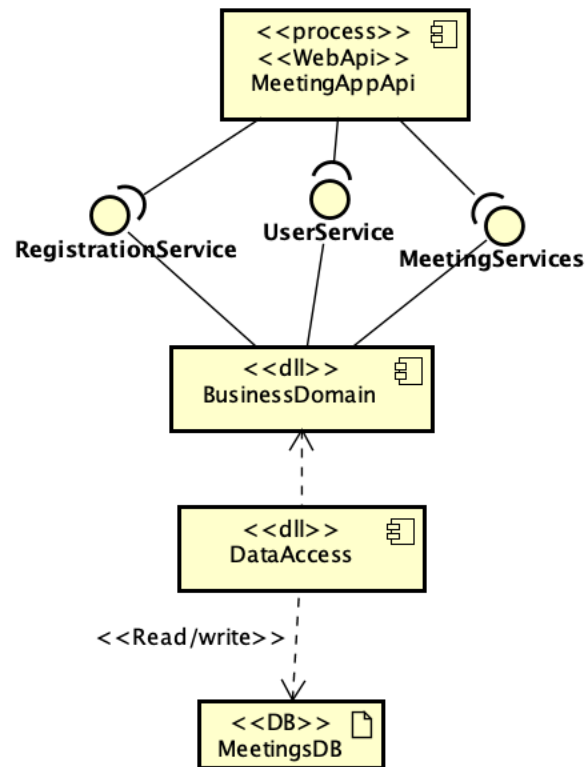
## Opción 1

### Módulos



- MeetingAppApi – Expone la API para acceder a las funcionalidades del backend.
- BusinessDomain – expone interfaces para que la API realice operaciones sobre las reglas del negocio. Contiene la lógica del negocio y los elementos del dominio e interactúa con DataAccess para operar sobre la persistencia.
- DataAccess – Maneja la tecnología de acceso a base de datos (ej. ORM).

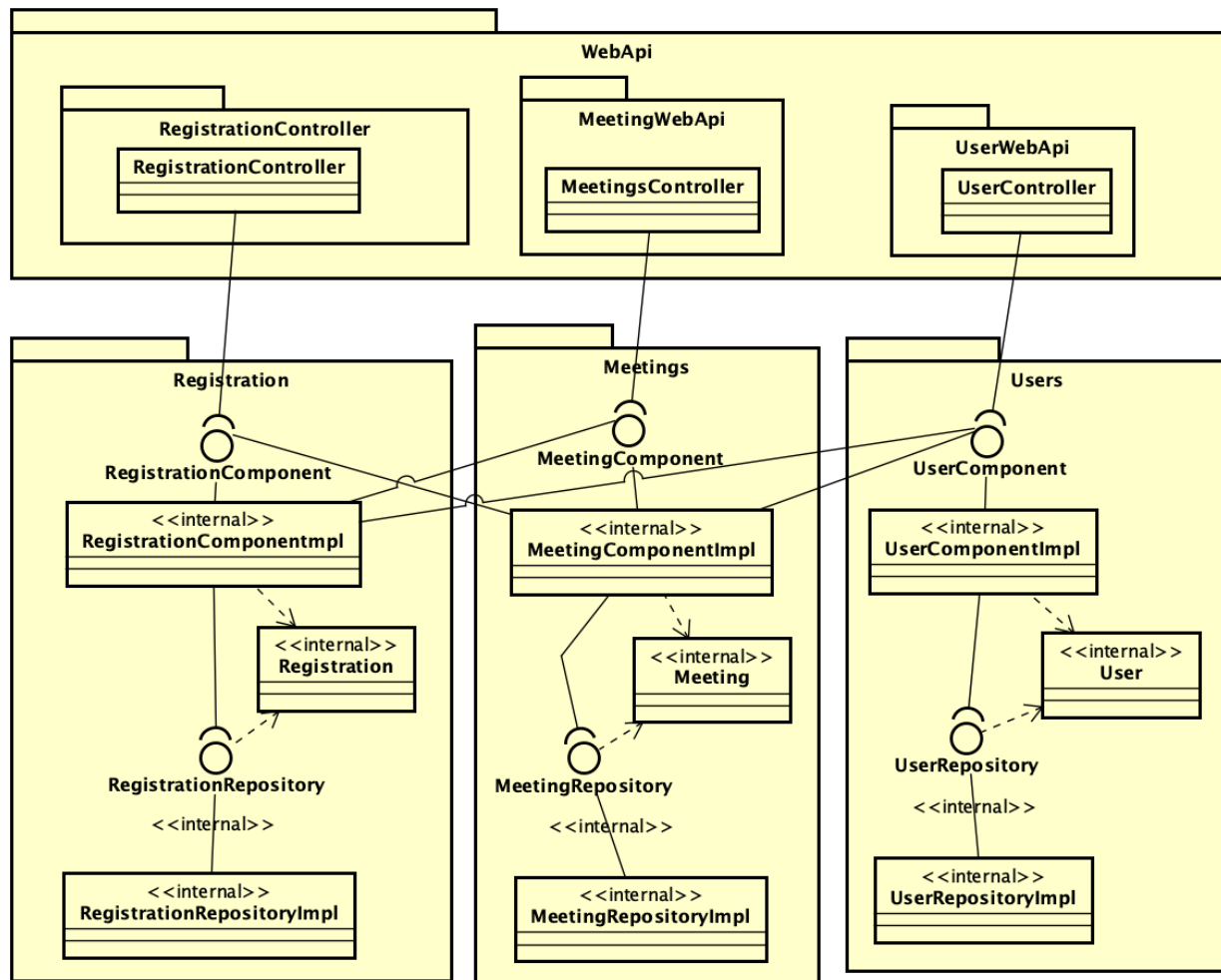
## Componentes y Conectores



- MeetingAppApi – proceso que implementa la API que expone la aplicación.
- BusinessDomain – dll que expone interfaces para que la API realice operaciones sobre las reglas del negocio. Contiene la lógica del negocio y los elementos del dominio. Accede mediante DataAccess a los datos.
- DataAccess – dll que maneja la tecnología de acceso a base de datos.
- MeetingsDB – base de datos que contiene la información que utiliza la app.

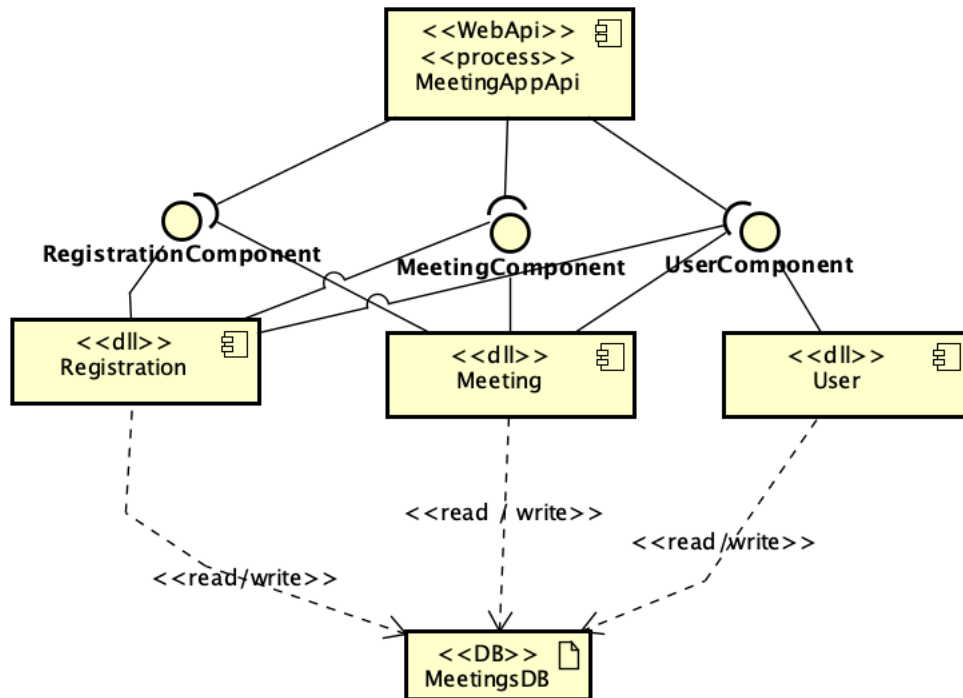
## Opción 2

### Módulos



- MeetingAppApi – Expone la API para acceder a las funcionalidades del backend.
- Registration - expone la lógica para las features de Registro a las reuniones, interactúa con Reuniones y Usuarios (ver interfaces) y accede a los datos de Registro.
- Meetings - expone la lógica para las features de las Reuniones, interactúa con Registro y Usuarios (ver interfaces) y accede a los datos de Reuniones.
- Users - expone la lógica para las features de los Usuarios y accede a los datos de Reuniones.

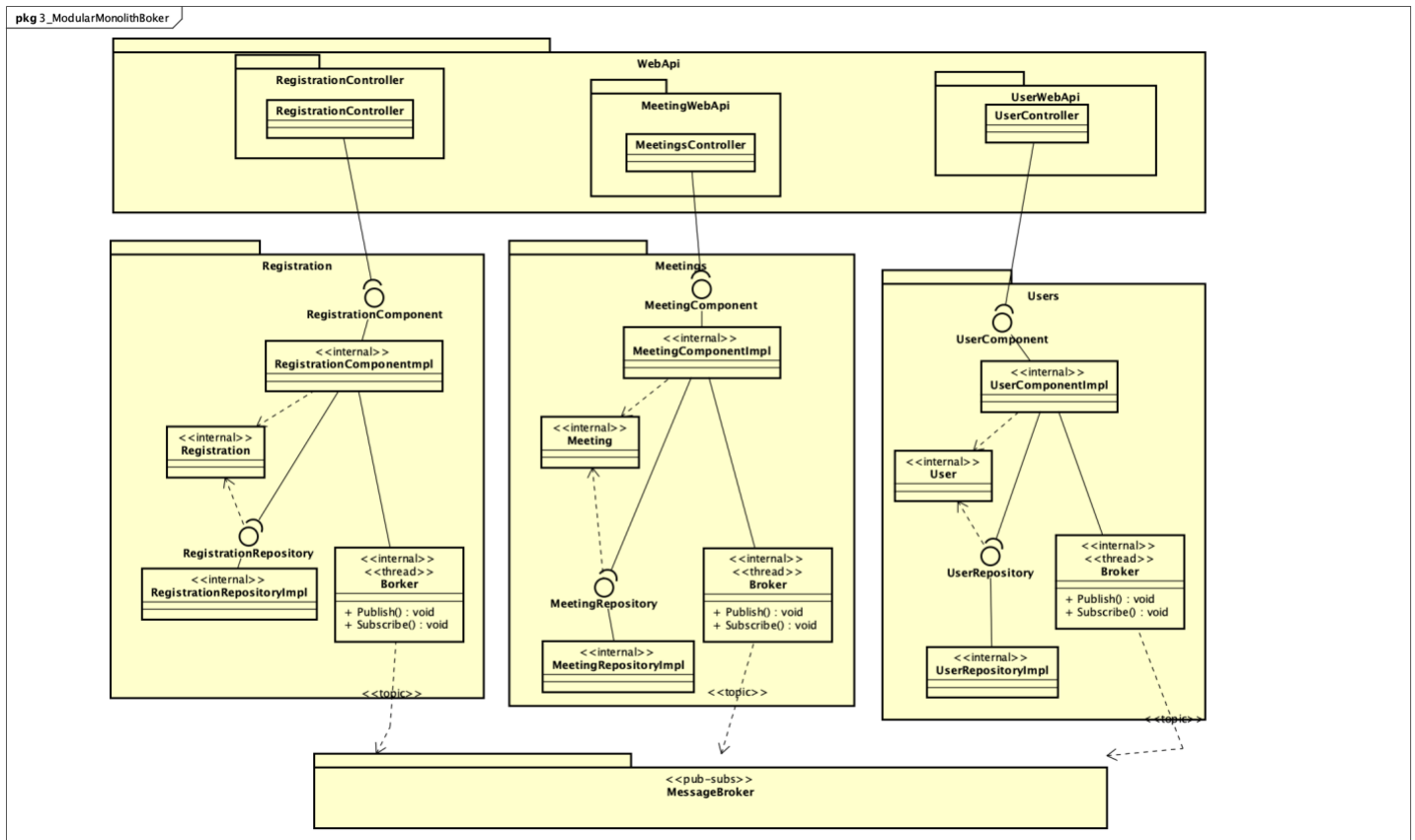
## Componentes y Conectores



- MeetingAppApi – proceso que expone la API (REST) para acceder a las funcionalidades del backend.
- Registration – dll que expone la lógica para las features de Registro a las reuniones, interactúa con Reuniones y Usuarios (ver interfaces) y accede a los datos de Registro.
- Meetings - dll que expone la lógica para las features de las Reuniones, interactúa con Registro y Usuarios (ver interfaces) y accede a los datos de Reuniones.
- Users - dll expone la lógica para las features de los Usuarios y accede a los datos de Reuniones.
- MeetingsDB – base de datos que contiene la información que utiliza la app.

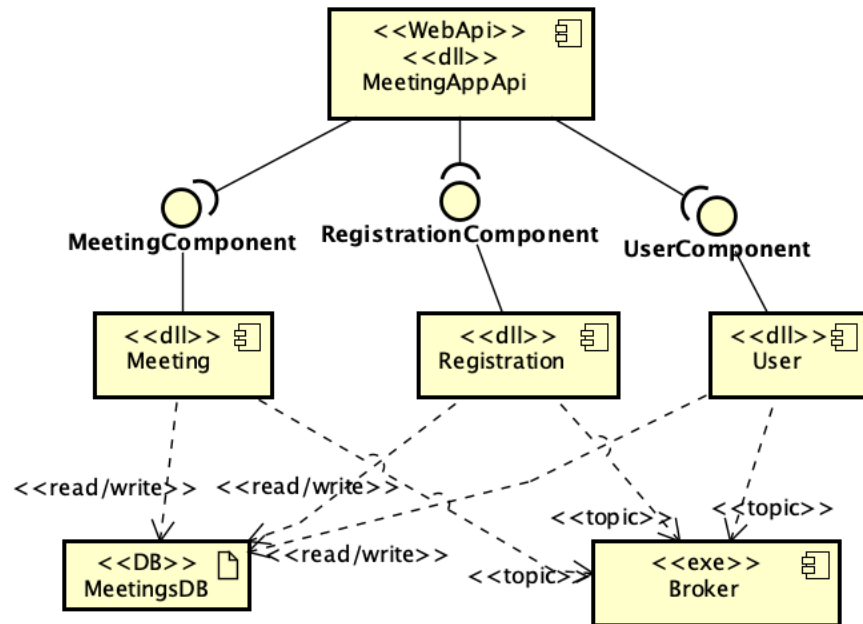
## Opción 3

### Módulos



- MeetingAppApi – Expone la API (REST) para acceder a las funcionalidades del backend.
- Registration - expone la lógica para las features de Registro a las reuniones, accede a los datos de Registro, y publica eventos asociados a las operaciones que se realiza, recibe eventos lanzados por otros componentes.
- Meetings - expone la lógica para las features de las Reuniones, accede a los datos de Reuniones y publica eventos asociados a las operaciones que se realiza, recibe eventos lanzados por otros componentes.
- Users - expone la lógica para las features de los Usuarios, accede a los datos de Reuniones y publica eventos asociados a las operaciones que se realiza, recibe eventos lanzados por otros componentes.

## Componentes y Conectores



- MeetingAppApi – proceso que expone la API para acceder a las funcionalidades del backend.
- Registration – dll que expone la lógica para las features de Registro a las reuniones, interactúa, accede a los datos de Registro y publica y reacciona eventos que llegan a través del broker.
- Meetings - dll que expone la lógica para las features de las Reuniones, accede a los datos de Reuniones y publica y reacciona eventos que llegan a través del broker.
- Users - dll expone la lógica para las features de los Usuarios, accede a los datos de Reuniones y publica y reacciona eventos que llegan a través del broker.
- MeetingsDB – base de datos que contiene la información que utiliza la app.
- Broker – proceso que implementa un Publicador – Subscriptor.