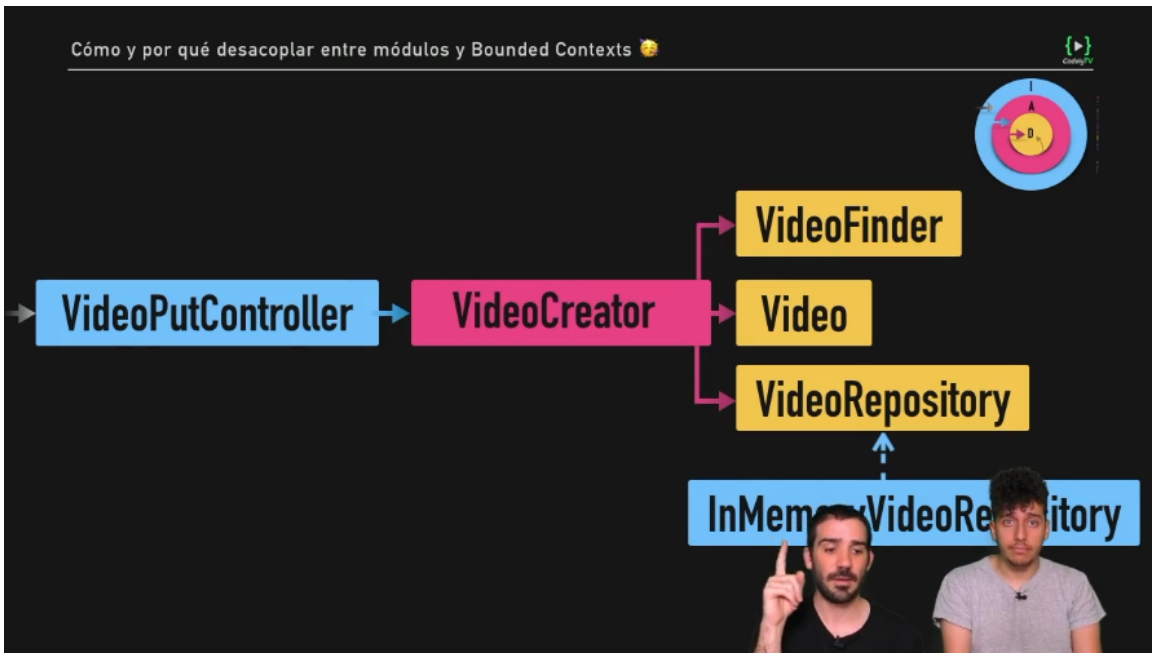
Watch: **Cómo y por qué desacoplar entre módulos y Bounded Contexts**

¿Cómo y por qué desacoplar entre Módulos y Bounded Contexts?

Flujo de petición



Utilizando el ejemplo de *Crear un Video* que venimos utilizando en otras lecciones, recordamos que desde fuera de nuestra aplicación llegaría una petición al controlador `VideoPutController` y desde aquí irá al servicio de aplicación `VideoCreator`, que interactúa con `VideoFinder` para comprobar si ya existe el video y finalmente con el repositorio `VideoRepository` para persistirlo en BD. En esta ocasión lo haríamos a través de una implementación `InMemoryVideoRepository` que almacenará los datos en un array en memoria

Estamos viendo que el caso de uso `VideoCreator` se va a encargar de guardar en base de datos, pero es posible que en un futuro queramos que cuando se cree un video se incremente un contador con el total de videos que hay, notificar a los subscriptores, enviar email de confirmación... En definitiva cada vez más acciones derivadas del hecho de crear un video, asumiendo más responsabilidades y rompiendo con el principio SRP de SOLID

¿Qué podemos hacer para evitar esto?

Una solución a este problema nos la ofrecen los **Eventos de Dominio**: nuestro VideoCreator lo que hará será publicar un evento de dominio en el exchange al que podrán subscribirse otros módulos independientes y realizar las las acciones pertinentes cada vez que lo reciban

Gracias a este flujo, si queremos añadir nuevas acciones derivadas de crear un nuevo vídeo, lo único que tendremos que hacer será añadir una nueva cola conectada al exchange sin necesidad de modificar el código existente

Como ya hemos comentado, aunque ahora nos encontremos con mayor número de clases en nuestra aplicación, estamos ganando en **independencia** de los distintos equipos de trabajo que puedan estar implicados. Por otro lado, una vez que nuestro caso de uso realiza un único cometido, la respuesta será mucho más rápida y mejoraremos en términos de **rendimiento**

Finalmente, al desarrollar de esta forma nuestra aplicación, estaremos ganando en **Modelado de datos**, siguiendo con el ejemplo que vimos al principio con el caso de **Curso**, recordemos que tenía el total de videos y el total de pasos gracias a que lo tenemos en el servicio de Video, lo tendremos modelado de forma óptima para poder recuperarlo directamente de BD sin tener que realizar ningún cálculo adicional

¿Alguna Duda?

¡Nos vemos en el siguiente video: Refactoring del caso de uso añadiendo publicación de eventos! Si tienes cualquier duda, sugerencias sobre este video o quieres compartir alguna otra propuesta que te interese puedes abrir una discusión nueva aquí abajo. 👇 👇