

Watch: [Refactoring del caso de uso añadiendo publicación de eventos](#)

Publicando evento en caso de uso Video Published 🎉

Clase VideoPublisher:

```
public final class VideoPublisher{  
  
    private final EventBus eventBus;  
  
    public VideoPublisher(EventBus eventBus) {  
  
        this.eventBus = eventBus;  
  
    }  
  
    public void publish(String rawTitle, String  
rawDescription) {  
  
        final var title = new VideoTitle(rawTitle);  
        final var description = new  
VideoDescription(rawDescription);  
  
        final var video = Video.publish(title, description);  
  
        eventBus.publish(video.pullDomainEvents());  
    }  
}
```

Clase Video:

```
public final class Video {  
  
    private VideoTitle title;  
    private VideoDescription description;  
  
    private Video(VideoTitle title, VideoDescription  
description) {
```

```
        this.title = title;
        this.description = description;
    }

    public static Video publish(VideoTitle title,
VideoDescription description) extends AggregateRoot {
        var video = new Video(title, description);

        var videoCreated = new VideoPublished(title.value(),
description.value());

        video.record(videoCreated);

        return video;
    }
}
```

Vemos en nuestro caso de uso VideoPublisher cómo el resultado será que va a publicarse un evento de dominio VideoPublished

- Evento de Dominio: Se trata de una acción que se ha realizado, lo cual se traduce en una mutación de estado en nuestro sistema

```
public final class VideoPublished implements DomainEvent {
    private static final String FULL_QUALIFIED_EVENT_NAME =
"codelytv.video.video.event.1.video.published";

    private final String title;
    private final String description;

    public VideoPublished(String title, String description) {
        this.title = title;
        this.description = description;
    }

    public String fullQualifiedEventName() {
        return FULL_QUALIFIED_EVENT_NAME;
    }

    public String title() {
        return title;
    }

    public String description() {
        return description;
    }
}
```

Como vemos, VideoPublished no deja ser un modelo, implementando la interface DomainEvent que nos obligará a implementar el método *fullQualifiedEventName* para devolver el nombre del evento

El hecho de que los atributos del evento de dominio se definan como escalares nos va a permitir por una parte que esté desacoplado de contextos o módulos, y por otra que sean fácilmente serializables

Asumiendo Responsabilidades

Tal como hemos visto, es el caso de uso tendrá las implicaciones de la atomicidad en términos de transaccionabilidad de BD y publicaciones de eventos, pero serán los eventos que recoja del **Aggregate Root**, por lo que será el agregado el responsable de generar los eventos

Ojo 🙄! El agregado no es quien publica el evento, sino que lo va a generar y registrar guardándoselo dentro

Clase AggregateRoot:

```
public abstract class AggregateRoot {
    private List<DomainEvent> recordedDomainEvents = new
LinkedList<>();

    final public List<DomainEvent> pullDomainEvents() {
        final var recordedDomainEvents =
this.recordedDomainEvents;
        this.recordedDomainEvents = new LinkedList<>();

        return recordedDomainEvents;
    }

    final protected void record(DomainEvent event) {
        recordedDomainEvents.add(event);
    }
}
```

Lo que nos va a permitir AggregateRoot es dotar a nuestras clases de la capacidad de registrar eventos. Así podremos desde el Application Service publicarlos a través de un publicador

Fijaos que el método se llama `pullDomainEvents`, y esto es así con toda la intencionalidad semántica, ya que en el cuerpo del método estaremos vaciando la lista de eventos que tiene la clase. De este modo, nos aseguramos de que si dentro del caso de uso seguimos realizando acciones con esa clase y volvemos a publicar eventos, no se repitan aquellos que ya se habían procesado antes

Aquí tenéis los repos con los ejemplos en PHP, Scala y Java, al cual pertenece el código que hemos visto en este video

- [php-ddd-example](#)
- [scala-ddd-example](#)
- [java-ddd-example](#)

¿Alguna Duda?

Si tienes cualquier duda o sugerencia relacionada con este video no dudes en abrir una nueva discusión aquí abajo 🙌 🙌 🙌