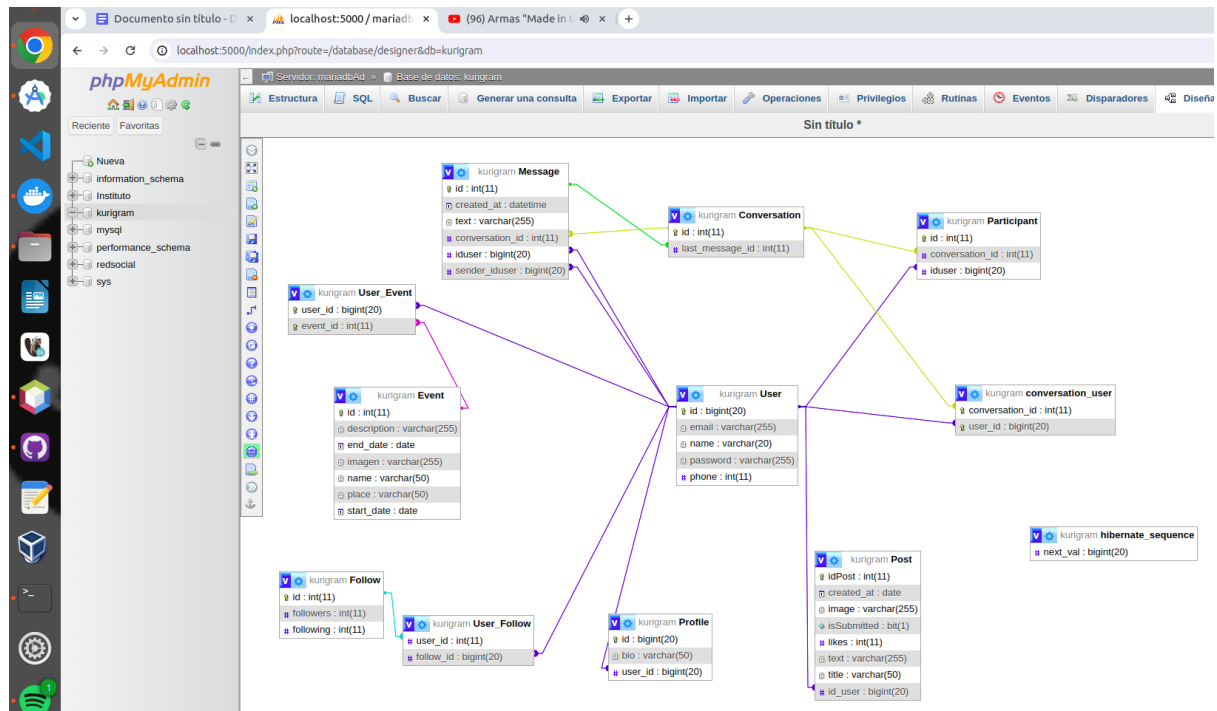


Raul Sales Herranz

## Diagrama E-R:



-Tabla User: pensando que es una bbdd de una red social, todo debe girar entorno a el usuario(tema mensajes, conversación, participantes, quizás con firebase seria la opcion perfecta para realizar un chat). Contamos con varias relaciones muchos a muchos, un usuario a muchas acciones, y uno a uno. El propio hibernate nos crea las tablas intermedias si así lo hemos especificado y configurado en el hibernate.cfg.xml. También crea una tabla suelta hibernate. Esta acción la realizan varios o ORM desde mi experiéncia, como Doctrine.

-Tabla Event: utilizada para trabajar los eventos de la red social. Se relaciona con los usuarios y almacena información de estos

-Tabla Messages: Diseñada para almacenar información de mensajes.

-Tabla Follow: Utilizada para que varios usuarios se puedan seguir entre sí. se relaciona a la tabla usuarios como muchos a muchos

-Tabla Profile: relación uno a uno con el usuario.

-Tabla Participant: Tabla diseñada para guardar los participantes de una funcionalidad de una conversación. (Podrian ser más de 2)

-Tabla Conversation; Se utiliza para guardar los datos de una conversacion entre varios participantes.

-Tabla Post: contiene información de los posts de los usuarios almacenando datos como su fecha, y los usuarios que lo han hecho

Se ha implementado o intentado hql para los findAll().

-MVC (Sin la Vista):

-> controllers: manejan la lógica de las funcionalidades a realizar.

-> persistence: no es buena práctica tener la lógica en los repositorios e incluso en main. por ello deberíamos desplazar la lógica a un repositorio. Pero tengo mis dudas, se que en Spring si se trabaja así.

[illegible]

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD 3.0//EN" "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
3
4 <hibernate-configuration>
5   <session-factory>
6     <property name="hibernate.connection.driver_class">com.mysql.cj.jdbc.Driver</property>
7     <property name="hibernate.connection.url">jdbc:mysql://localhost:3308/kurigram</property>
8     <property name="connection.username">root</property>
9     <property name="connection.password">root</property>
10    <property name="connection.pool_size">5</property>
11    <property name="dialect">org.hibernate.dialect.MySQL5InnoDBDialect</property>
12    <property name="current_session_context_class">thread</property>
13    <property name="show_sql">true</property>
14    <property name="hbm2ddl.auto">update</property>
15    <mapping class="projecte.entities.User" />
16    <mapping class="projecte.entities.Event" />
17    <mapping class="projecte.entities.Posts" />
18    <mapping class="projecte.entities.Message" />
19    <mapping class="projecte.entities.Follow" />
20    <mapping class="projecte.entities.Conversation" />
21    <mapping class="projecte.entities.Participant" />
22    <mapping class="projecte.entities.Profile" />
23
24  </session-factory>
25 </hibernate-configuration>

```

-Ejecucion:  
Aquí podríamos acceder a una entidad y realizar operaciones sobre ella, como los usuarios, listarlos, borrarlos, actualizarlos...

```
pc-raul@pc-raul-ROG-Strix-G512LV-G512LV: ~/2DAM_CourseSalesRaul/Uni...
(base) pc-raul@pc-raul-ROG-Strix-G512LV-G512LV:~/2DAM_CourseSalesRaul/Unitat3/pr
jecte$ ./gradlew run --console=plain
Path for java installation '/usr/lib/jvm/openjdk-11' (Common Linux Locations) do
es not contain a java executable
Path for java installation '/usr/lib/jvm/openjdk-17' (Common Linux Locations) do
es not contain a java executable
> Task :app:compileJava UP-TO-DATE
> Task :app:processResources NO-SOURCE
> Task :app:classes UP-TO-DATE

> Task :app:run
Seleccione una opción:
1. Users
2. Events
3. Messages
4. Posts
5. Profiles
0. Salir

```