

Trabajo Práctico Especial

# MakeMymeeting

(Entrega 1)

Autores:

Colamai, Daniela

Rosenfeld, Federico Nahuel

Facultad de Cs. Exactas

Universidad Nacional del Centro de la Provincia de Buenos Aires

Cátedra Arquitecturas WEB

## Resumen

**MakeMyMeeting** es una aplicación para programar reuniones de trabajo o actividades de un usuario, compartirlas, y poder coordinarlas con otros usuarios.

En este Informe se especificarán la toma de decisiones ,en calidad de estructuras y funcionalidades , utilizadas para lograr el desarrollo de la aplicación.

## Repositorio

<https://github.com/elmuddy/ArquitecturasWebTPE/tree/pruebas>

## Entidades

Para el desarrollo de la aplicación se generaron 4 Entidades POJO.

- Usuario
- Actividad
- Sala
- Calendario

Con las cuales se formó un modelo de Agenda.

## Descripciones

### **Usuario:**

Esta entidad es la encargada de manejar los datos nombre y apellido de los Usuarios, el conjunto de calendarios, propios y compartidos, y actividades pendientes que estos poseen.

### **Actividad:**

Esta entidad es la encargada de mantener un registro de información sobre la reunión a ejecutar, consiguiendo avisar a los Invitados, los cuales los almacena en un registro de pendientes hasta que estos decidan aceptar la invitación. Para mantener una actividad valida esta genera una reserva en la sala designada evitando la posibilidad de que haya superposiciones.

### **Sala:**

Esta entidad es la poseedora de la informacion de la localización en donde se realizara una actividad. A su vez , posee un listado de actividades que la utilizan para evitar que esta permita una superposición.

### **Calendario:**

Esta entidad es la encargada de tener un registro de actividades asociadas a este mismo. Un identificador de quien es el creador de dicho calendario.

# Servicios

## **DAOSala:**

Esta clase posee el conjunto de métodos asociados con la entidad Sala

- crearSala (...): Crea una nueva sala utilizando el nombre y dirección pasados por parámetro.
- getSala(...): Regresa una Sala desde la Base de Datos utilizando un identificador ID pasado por parámetro.

## **DAOUsuario:**

Esta clase posee el conjunto de métodos asociados con la entidad usuario

- crearUsuario(...): Dado un conjunto de valores por parámetro genera un Usuario
- getUsers(...): Regresa un listado con todos los Usuarios de la Base de Datos
- getUser(...): Dado un IdUsuario por parámetro, regresa ese usuario desde la Base de Datos.
- getInfoUsuario(...): Dado un IdUsuario por parámetro, se muestra la información usuario desde la Base de Datos.
- getActividadDeUsuario(...): En base a una Fecha Dada por parametro, muestra un listado de Actividades que posee ese Usuario
- getActividadDeUsuarioEntreDias(...): En base a dos Fechas Dadas por parametro, muestra un listado de Actividades que posee ese Usuario entre esas fechas.

**DAOCalendario:**

Esta clase posee el conjunto de métodos asociados con la entidad Calendario

- crearCalendario(...): Dado un conjunto de valores por parámetro genera un Calendario
- getCalendarioXid(...): Dado un id de un calendario por parámetro regresa ese Calendario desde la Base de Datos.
- getCalendarioXusuario(...): Dado un IdUsuario por parámetro, regresa ese usuario desde la Base de Datos.

**DAOActividad:**

Esta clase posee el conjunto de métodos asociados con la entidad Actividad

- crearActividad(...): Dado un conjunto de valores por parámetro genera una Actividad
- getActividades(...): Regresa un listado con todas las actividades de la Base de Datos
- getActividadesUsuario(...): Dado un IdUsuario por parámetro, regresa las actividades relacionadas a ese usuario desde la Base de Datos.
- getActividadesSobrepuestas(...) Dado un idUsuario y una actividad, muestra un listado de actividades superpuestas en horarios con esta misma.

## Funcionalidades Pedidas

- a. Inicializar la base, si no está inicializada.

Se resuelve en : META-INF/persistence.xml

- b. Consultar usuarios y calendarios. En particular se debe probar

- i. Consulta de todos los datos de un usuario.

*Se resuelve en :* getInfoUsuario(...) en DAOUsuario

- ii. Dado un usuario y un día, retornar todas sus reuniones.

*Se resuelve en :* getActividadDeUsuario(...) en DAOUsuario

- iii. Dado un usuario y un rango de fechas, retornar todas sus reuniones.

*Se resuelve en :* getActividadDeUsuarioEntreDias(...) en DAOUsuario

- iv. Dado un usuario y una nueva reunión, retornar todas las reuniones que se superpongan con la nueva reunión.

*Se resuelve en :* getActividadesSobrepuestas(...) en DAOActividad

## Aclaraciones

### **General:**

Se cambio todo el modelo de la aplicación y se volvieron las Entidades POJO, desplazando las funcionalidades a sus respectivos DAO.

### **Actividad:**

1) En la Entidad actividad se opto por la utilización de 2 listas de usuarios.

-Pendientes

-Invitados

Los Usuarios de la lista de pendientes son aquellos que fueron invitados a agendarse dicha Actividad, pero aun no lo han hecho.

Los Usuarios de la lista de Invitados son aquellos que fueron invitados para agendarse a dicha Actividad y aceptaron la invitación. Al momento de hacerlo, dejaron de estar en la lista de Pendientes y pasaron a la de Invitados.

2) Es responsabilidad de la Actividad el notificar la situación de Pendiente de un Usuario invitado a participar de la misma.