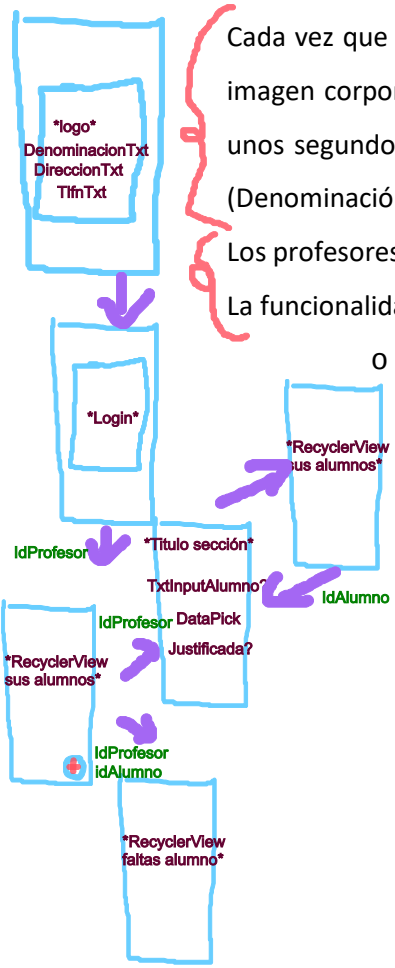
Ejercicio 7 Bases de datos

Un centro educativo nos ha pedido que le realicemos una app para Gestionar las faltas de asistencia de sus alumnos.



Cada vez que se ejecuta la aplicación se mostrará una pantalla de bienvenida que contiene la imagen corporativa del Centro y los datos de Centro. Esta información desaparecerá pasados unos segundos. La información del Centro se encuentra almacenada en el fichero centro.xml (Denominación, Dirección, Teléfono), dicho fichero estará en la carpeta de recursos.

Los profesores se deberán validar para poder acceder al resto de opciones.

La funcionalidad de la que dispondrá será la siguiente:

- **Faltas.** Introducir las faltas de un alumno un día y hora determinada. En el caso de que el alumno ya tuviera la falta introducida se deberá completar en el campo observaciones con el nombre del profesor que ha intentado poner la falta, en el caso de que fuera distinto. Si el profesor es el mismo simplemente se le informará de que la falta ya estaba introducida. En el caso de no estuviera puesta la falta el campo observaciones se dejará en blanco.

  De forma automática se cogerá el profesor que lo ha intentado y se guardará en observaciones la id del profesor actual y se regresará

  Si es el mismo que ha intentado poner la falta, sino: Observaciones+=", profesorNuevo"

  Si todo bien se vuleve se crea la falta y volvemos atras

  Consulta de Alumnos. Se mostrará el código y nombre de los alumnos a los que un profesor imparte docencia.

  RecyclerView

  Necesario metodo getAlumnos(idProfesor)

  Cuando se realiza una pulsación corta sobre un alumno se mostrará las faltas de ese alumno, Fecha, Hora y si está justificada o no.

  RecyclerView

  Las faltas podrán ser justificadas (en el caso de que no lo estuvieran), siempre y cuando la falta hay sido puesta por el profesor logeado.

  Click largo supongo si el profesor es el mismo

Para ello se va a disponer de la base de datos en SQLLite CENTRO que contiene las siguientes tablas:

Class bd

getAlumnos(idProfesor): mutableList<Alumno>
getFaltas(idAlumno): mutableList<Falta>
getAllAlumnos(): mutableList<Alumno>

Tabla ALUMNOS

Código Alumno	Campo autonumérico. Clave Principal. (Código del alumno)
Nombre	Texto (Requerido)

En esta tabla se encuentran todos alumnos del Centro.

Tabla FALTAS-ALUMNO

Código	Campo autonumérico. Clave Principal.
Código alumno	Numérico (Requerido)

Fecha	Texto (Requerido)
Hora	Numérico Entero (Valor del 1-6) (Requerido)
Código Profesor	Numérico Entero (Requerido)
Justificada	booleano.
Observaciones	Texto

No puede haber dos registros del mismo alumno en una fecha y hora determinada.

Tabla PROFESOR

DNI Profesor	Texto. Clave Principal. Código del Profesor
login	Texto
contraseña	Texto
Nombre del profesor	Texto

No puede haber dos profesores con el mismo login y contraseña.

Tabla PROFESOR-ALUMNO

código	Campo autonumérico. Clave Principal.
Código Alumno	**Numérico Entero**
Código Profesor	**Numérico Entero**

Un profesor puede impartir docencia a varios alumnos

Notas:

- Se debe de crear la base de datos y dotarla de la información necesaria para poder comprobar el ejercicio.
- Se valorará la búsqueda de información para la resolución del ejercicio, como por ejemplo la utilización del DatePicker para introducir las fechas, la utilización de los cuadros de diálogo para mostrar los mensajes, ..
- El nombre de la base de datos será vuestro **Ut7Ej7PrimerApellidoNombre**
- El nombre del proyecto será **Ut7Ej7PrimerApellidoNombre**.
- El nombre del archivo comprimido será el mismo (se debe de subir la base de datos más el proyecto)
- Se deberá subir a la tarea creada en classroom. La fecha tope de subida es: 23 de enero de 2023.

Ejercicio 8 Bases de Datos

Realizar una app para Gestionar los eventos de una Comunidad de Programadores.

LayoutInicio

Cada vez que se ejecuta la aplicación se mostrará una pantalla de bienvenida que contiene la imagen corporativa del Comunidad, una breve descripción de la aplicación, el autor y el año de la creación de la app. Esta desaparecerá después de unos segundos

LayoutLogin

La información de los usuarios se proporciona en el archivo de recursos "usuarios.xml" que deberá cargarse en el arranque la primera vez que se ejecute la aplicación. Los datos de los usuarios se guardarán en la tabla correspondiente. A continuación, se pedirá al usuario que se valide, introduciendo su login y contraseña.

Que se meta en un mutable nada más arrancar, y luego en la base de datos

cargarUsuarios():MutableList<Usuario>

Una vez que el usuario ha entrado en la App dispondrá de las siguientes opciones, dependiendo del tipo de perfil:

Diferentes tipos?

LayoutAltaEvento

- **Alta de Eventos.** Cualquier usuario de la aplicación, con perfil administrador, podrá dar de alta eventos.

  Se deberá de controlar que la fecha y la hora del evento no haya pasado y que no puede haber dos eventos simultáneamente.

  Comprobar que no se repita y un datapicker

  .

LayoutEventos

- **Consulta de Eventos.** Se mostrará solamente aquellos eventos que todavía no han sido. Estos eventos aparecerán ordenados por fecha y hora. Como coño ordeno esto?

  Cuando el usuario realice una pulsación larga sobre la tarjeta de evento se mostrará toda su información pudiéndose inscribirse a dicho evento o bien anular dicha inscripción en el caso de que ya se hubiera apuntado.

  Esta opción la podrán realizar todos los usuarios.

LayoutModifEvento

- **Modificación Eventos.** Los usuarios con perfil administradores podrán modificar el día y la hora del evento, siempre y cuando no haya pasado, teniendo en cuenta las mismas restricciones que para el alta.

Para ello se va a disponer de la base de datos en SQLLite **PROGRAMADORES** que contiene las siguientes tablas:

Tabla EVENTOS

id Campo autonumérico. Clave Principal.

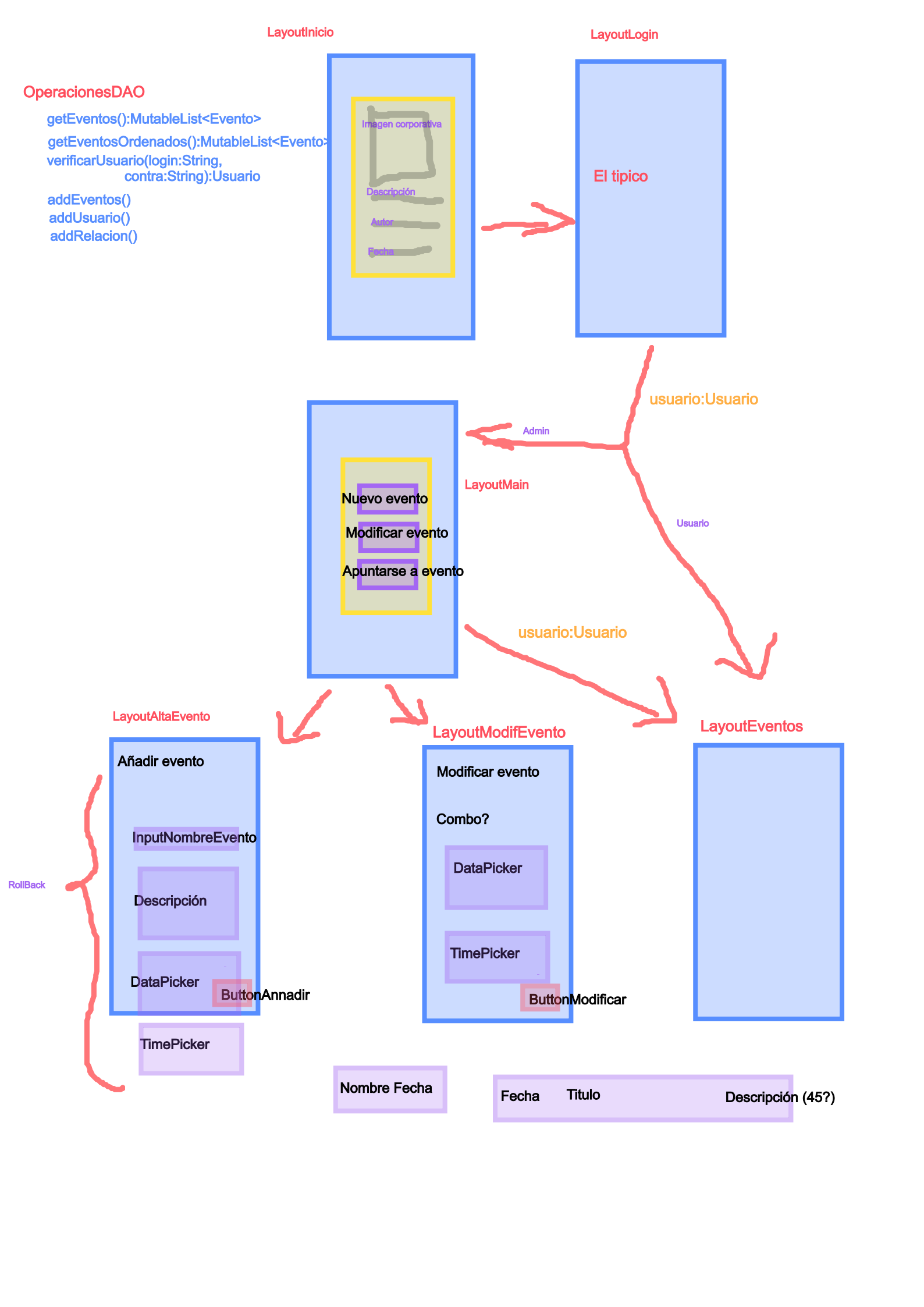

LayoutInicio
LayoutLogin
OperacionesDAO
getEventos():MutableList<Evento>
getEventosOrdenados():MutableList<Evento>
verificarUsuario(login:String,
contra:String):Usuario
addEventos()
addUsuario()
addRelacion()
Imagen corporativa
Descripción
Autor
Fecha
El tipico
usuario:Usuario
Admin
LayoutMain
Nuevo evento
Modificar evento
Apuntarse a evento
Usuario
usuario:Usuario
LayoutAltaEvento
LayoutModifEvento
LayoutEventos
Añadir evento
InputNombreEvento
Descripción
DataPicker
ButtonAnnadir
TimePicker
RollBack
Modificar evento
Combo?
DataPicker
TimePicker
ButtonModificar
Nombre Fecha
Fecha
Titulo
Descripción (45?)

Tabla EVENTOS

id	Campo autonumérico. Clave Principal.
Fecha evento	Texto (DD/MM/AAAA)
Hora	Texto (HH:MM)
Titulo	Texto
Descripción	Texto

En esta tabla se encuentran todos los eventos programados.

Tabla EVENTOS-USUARIO

_id	Campo autonumérico. Clave Principal.
Id_usuario	Campo numérico. Código del usuario
Id_evento	campo numérico. Código evento.

En esta tabla están los eventos a los que se ha apuntado el usuario

Tabla USUARIO

_id	Campo autonumérico. Clave Principal.
login	Texto
contrasena	Texto
Perfil	Texto A-> administrador U-> Usuario estándar.

Notas:

- Se debe de crear la base de datos y dotarla de la información necesaria para poder comprobar el ejercicio.
- Se valorará la búsqueda de información para la resolución del ejercicio, como por ejemplo la utilización del DatePicker para introducir las fechas, la utilización de los cuadros de diálogo para mostrar los mensajes, …..
- El nombre de la base de datos será vuestro **Ut7Ej8PrimerApellidoNombre**
- El nombre del proyecto será **Ut7Ej8PrimerApellidoNombre**.
- El nombre del archivo comprimido será el mismo (se debe de subir la base de datos más el proyecto)
- Se deberá subir a la tarea creada en classroom. La fecha tope de subida es: 23 de enero de 2023.